



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Desarrollo de una herramienta que permita la migración de
datos del formato NI-TDMS a

Trabajo Fin de Grado

Grado en Tecnologías Interactivas

AUTOR/A: Calabuig Artes, Joan

Tutor/a: Marín-Roig Ramón, José

Cotutor/a externo: GARCIA GARCIA, MARCIAL

CURSO ACADÉMICO: 2021/2022

UNIVERSITAT POLITÈCNICA DE VAÈNCIA

ESCOLA POLITÈCNICA SUPERIOR DE GANDIA

GRADO EN TECNOLOGIAS INTERACTIVAS



**UNIVERSITAT
POLITÈCNICA
DE VALENCIA**



**ESCOLA POLITÈCNICA
SUPERIOR DE GANDIA**

**DESARROLLO DE UNA HERRAMIENTA QUE
PERMITA MIGRAR LOS DATOS DEL FORMATO NI-
TDMS A INFLUXDB**

TRABAJO FINAL DE GRADO

AUTOR:

JOAN CALABUIG ARTES

TUORES:

JOSE MARÍN-ROIG RAMÓN

MARCIAL GARCÍA GARCÍA

GANDIA, 29 de JUNIO de 2022



Resumen

Los datos nos abordan hoy en día, se genera información de cualquier objeto o dispositivo y esto plantea retos importantes ante la capacidad de almacenamiento o procesado de estos datos.

En este proyecto se aborda la labor de actualizar y migrar un sistema de un formato de archivos a otro formato de almacenamiento con base de datos de series temporales. Esto supone la actualización de las aplicaciones actuales y desarrollo de otras nuevas para hacer la migración lo más fácil y segura posible.

Al final del proyecto se observan los resultados que demuestran las mejoras de funcionamiento.

Summary

The data we handle today, information is generated from any object or device and this poses major challenges in the storage or processing capacity of this data.

This project addresses the task of updating and migrating a system from one file format to another time series database storage format. This involves updating current applications and developing new ones to make migration as easy and secure as possible.

At the end of the project, the results of the operational improvements can be seen.

Palabras clave

Influx DB, LabVIEW, TDMS, Flux





Agradecimientos

La familia es parte fundamental y necesaria de toda persona, por ese motivo mi mayor agradecimiento va para los míos.

Agradecer a mis tutores, tanto académico como de prácticas laborales, José Marín-Roig y Marcial García por la ayuda y el apoyo para realizar este proyecto.

Agradecer a la empresa AUTIS INGENIEROS por la confianza y la oportunidad de realizar este proyecto en colaboración con su equipo.

Por último, agradecerme a mí mismo la constancia y perseverancia para alcanzar mis metas, las cuales me hacen mejorar cada día.





Índice

1	Introducción	6
1.1	Motivación	7
1.2	Objetivos y plan de acción	8
1.3	Impacto esperado	9
1.4	Metodología	10
1.5	Estructura	11
2	Estado del arte	12
3	Análisis del problema	16
3.1	Análisis de soluciones.....	16
3.2	Solución adoptada.....	16
4	Diseño de la solución.....	17
4.1	Arquitectura del sistema	17
4.2	Diseño detallado	18
4.3	Tecnologías utilizadas.....	20
5	Desarrollo	21
5.1	Aprendizaje de las tecnologías.....	21
5.1.1	LabVIEW	21
5.1.2	TDMS	25
5.1.3	InfluxDB	26
5.2	Instalación y puesta en marcha de InfluxDB	28
5.2.1	Instalación del servicio de InfluxDB.....	28
5.2.2	Acceso a la web cliente	30
5.3	Pruebas de funcionamiento	32
5.3.1	Escritura de datos.....	32
5.3.2	Consulta de datos.....	33
5.3.3	Desarrollo de la API de pruebas y aprendizaje.....	35
5.4	Desarrollo de aplicaciones de migración	38
5.4.1	Aplicación de atributos.....	38
5.4.2	Aplicación de migración	39
5.4.3	Actualización de la aplicación general	40
5.5	Desarrollo de scripts de consulta con Flux.....	41
6	Implantación y testeo.....	43
7	Conclusiones y trabajo futuro	47
8	Bibliografía	48





Índice figuras

Figura 1 Evolución de las tecnologías de la industria.....	12
Figura 2 Ámbitos de desarrollo de IIoT	13
Figura 3 Sistema utilizado por la empresa Vleemo.....	15
Figura 4 Arquitectura del sistema	17
Figura 5 Leyenda arquitectura del sistema	17
Figura 6 APP_Influx_Set_TDMS_Atributes.vi.....	18
Figura 7 App_TDMS_to_influx_queries.vi.....	18
Figura 8 Formato de entrada de datos.....	19
Figura 9 Ejemplo de la ventana Project Explorer	22
Figura 10 Ejemplo de Panel de Conexión.....	23
Figura 11 Ejemplo de Front Panel	23
Figura 12 Ejemplo de código en LabVIEW.....	24
Figura 13 Paleta de Toolkits y librerías.....	24
Figura 14 Estructura interna de ficheros TDMS.....	25
Figura 15 Funcionalidades y características de InfluxDB	27
Figura 16 Creación del servicio de InfluxDB.....	29
Figura 17 Configuración de usuario y organización	30
Figura 18 Vista sobre la información de la organización	31
Figura 19 Sección para obtener el token de acceso.....	31
Figura 20 Crear un bucket de almacenamiento	32
Figura 21 Formato de escritura de datos.....	32
Figura 22 Consulta con el constructor de consultas	33
Figura 23 Consulta con Flux	34
Figura 24 Visualización de datos en gráficos.....	34
Figura 25 Visualización de datos en tabla	34
Figura 26 Consulta con Flux desde Postman.....	35
Figura 27 Respuesta desde InfluxDB	35
Figura 28 Ventana de la aplicación de LabVIEW para InfluxDB	36
Figura 29 Ejemplo de código de la aplicación	36
Figura 30 Diagrama de flujo de la aplicación	37
Figura 31 Antes y después de añadir los nuevos atributos.....	38
Figura 32 Sección de gráficos de históricos	43
Figura 33 Sección de gráficos de históricos	44
Figura 34 Sección de gráficos de inversores	45
Figura 35 Historiales diarios de energía generada.....	45
Figura 36 Mapa de calor de los datos de producción diaria	46





1 Introducció

Hoy en día, todo dispositivo equipado mínimamente con electrónica produce datos y siempre que se habla de ellos, se piensa en su procesado y en el almacenamiento seguro. Pero hay otra parte importante, y es la accesibilidad a esos datos y la información que pueden proporcionar.

Por este motivo en este trabajo de fin de grado se desarrolla una herramienta, en colaboración con la empresa AUTIS INGENIEROS, que permite migrar el tipo de almacenamiento de datos del formato NI-TDMS a la base de datos de series temporales InfluxDB, consiguiendo una mejora en los tiempos de acceso y procesamiento de los datos.

Actualmente la empresa dispone de un producto software llamado PSFVIEW para la monitorización y registro de datos e históricos de parques fotovoltaicos. Con la antigua implementación, los datos se almacenan en archivos de formato TDMS, formato desarrollado por NI (Nacional Instruments) con gran eficiencia en desarrollos software LabVIEW.

Debido a la migración de las aplicaciones de monitorización al entorno web, se abren nuevas posibilidades de acceso simultaneo y exportación masiva de datos y ahí es donde entra InfluxDB.

El funcionamiento previo a la nueva implementación consiste en la adquisición de datos de sensores e inversores ubicados en los parques, a partir de aplicaciones que adquieren y procesan los datos de los sensores y una vez procesados se almacenan en archivos TDMS. Cuando se desea consultar los datos, la web realiza la petición al Webservice desarrollado en LabVIEW, consulta los datos de los archivos TDMS y devuelve el resultado al cliente.

Con la utilización de esta base de datos de series temporales, se pretende mejorar el rendimiento del producto PSFVIEW y adaptar el funcionamiento a las nuevas integraciones web con servidor REST desarrollado en NodeJS desde el cual se accede a los datos almacenados en el servidor de InfluxDB.

La herramienta se desarrolla con LabVIEW, para permitir migrar los histórico y también los nuevos registros.





1.1 Motivación

La realización de este proyecto viene motivada por el deseo de mejorar las prestaciones del producto PSFVIEW y ofrecer mejor servicio, así como, centralizar los datos en un sistema con un tipo de estructura de almacenamiento diseñado para la colección de datos a monitorizar.

La idea surgió al conocer la base de datos de series temporales InfluxDB y el potencial que tiene para proyectos como este, permitiendo mejorar la forma y los tiempos de accesibilidad. InfluxDB presenta la posibilidad de centralizar los datos con la información en un mismo sistema, a diferencia de TDMS que organiza la información en conjuntos de archivos lo que puede causar pérdida de información fácilmente al extraviar un archivo de la colección.

Después del análisis de InfluxDB y comprobar que es una solución que se adapta perfectamente al producto PSFVIEW, se decide realizar la migración a este sistema de almacenamiento que conlleva el desarrollo de las aplicaciones de migración de los datos históricos y los generados en tiempo real.

A nivel personal, este proyecto ofrece una visión y experiencia en la implementación de bases de datos de series temporales, las cuales están en gran auge debido a la evolución de las tecnologías y los servicios, que cada día ofrecen más información del entorno, donde esa información se puede almacenar de forma muy sencilla y accesible en este tipo de base de datos.



1.2 Objetivos y plan de acción

La finalidad principal de este proyecto se describe como la migración del formato de almacenamiento de los datos en NI-TDMS, utilizado por el sistema de aplicaciones actual, a un formato de base de datos de series temporales como es InfluxDB.

Actualmente todo el sistema de adquisición y procesamiento de datos de los inversores y otros sensores ubicados en los parques fotovoltaicos se realiza con aplicaciones LabVIEW. Estas mismas aplicaciones almacenan los datos en el formato TDMS. Posteriormente, para la visualización de los datos vía web, se necesitan aplicaciones desarrolladas en LabVIEW para leer los datos almacenados en TDMS, estos datos eran consultados por los Webservice y de estos, enviados a las webs de visualización.

La nueva implementación del sistema de monitorización pretende que las aplicaciones de adquisición de los datos, desarrolladas en LabVIEW, se encarguen de almacenar los datos en las bases de datos de InfluxDB. Una vez los datos en el nuevo sistema de almacenamiento se desarrolla un servidor en NodeJS encargado de consultar los datos a los servicios de InfluxDB y responder con esos datos a las peticiones realizadas por la web cliente.

Para llevar a cabo de forma eficaz el objetivo del proyecto, se necesitan aplicaciones que realicen la adaptación y migrado de los datos históricos y aplicaciones que almacenen los datos generados en tiempo real en la nueva base de datos. Por tanto, desglosa este objetivo principal en partes más pequeñas que permiten realizar el trabajo por tramos y así poder asegurar el correcto funcionamiento del conjunto de aplicaciones una vez unidas todas ellas.

A continuación, se detalla el plan a seguir:

- ▶ Aprendizaje del entorno de desarrollo LabVIEW y su lenguaje gráfico.
- ▶ Aprendizaje de estructuras TDMS y forma de almacenamiento en InfluxDB.
- ▶ Diseño y desarrollo de aplicación para adaptar formato de archivos TDMS al nuevo formato InfluxDB.
- ▶ Diseño y desarrollo de aplicaciones para la lectura y almacenamiento de los datos en tiempo real en la nueva base de datos InfluxDB.
- ▶ Implementación y testeo de las aplicaciones de almacenamiento de datos en tiempo real.
- ▶ Implementación y testeo de las aplicaciones de migración de datos históricos.
- ▶ Puesta en producción de las aplicaciones.

Con la implementación de los objetivos mencionados, se considera completado el proceso de migración, al nuevo formato de almacenamiento, con éxito.



1.3 Impacto esperado

Al final del desarrollo detallado en este documento, se espera una mejora notable en el producto software PSFVIEW, en niveles de accesibilidad de la información al cliente.

También se espera una mejora en la manera de administrar la información y evitar pérdidas que causen retrasos y errores en recuperar la información de backup.

Gracias al procesado de datos que permite el sistema InfluxDB, también se espera una descarga de trabajo en el servidor, proporcionando una mejora de rendimiento en y tiempo de respuesta.



1.4 Metodología

Inicialmente y como en todo proyecto, se necesita una fase de investigación o aprendizaje sobre los temas a abordar. LabVIEW es una herramienta potente a la vez que compleja y por tanto la misma empresa desarrolladora proporciona cursos y documentación a sus usuarios. Para llevar a cabo el proyecto se han realizado dos cursos de formación, supervisados por la empresa AUTIS, llamados CORE 1 y CORE 2, posteriormente y como adaptación a la metodología de trabajo de la empresa se han realizado diferentes ejercicios utilizando la arquitectura AUTIS, arquitectura desarrollada en LabVIEW para uso interno de la empresa, todo ello supervisado también por la empresa AUTIS. La base de conocimientos adquiridos en este proceso de aprendizaje de la herramienta LabVIEW se consideran suficientes para realizar las implementaciones necesarias.

Parte del aprendizaje de la herramienta LabVIEW consiste en aprender la manipulación de archivos TDMS. Puesto que en este proyecto su tratamiento es de gran importancia, se profundiza en ello y en como modificarlos para preparar los datos internos de forma que se adapten de la mejor manera posible al formato de almacenamiento con los nuevos “Tags” o etiquetas de InfluxDB.

El aprendizaje de InfluxDB es más importante si cabe, pues es el destino de la migración de los datos y por tanto no solo se deben conocer las características para el traspaso de información, sino que también las características de funcionamiento y administración para el futuro correcto funcionamiento. Para ello se profundiza en el aprendizaje también de Flux, el lenguaje de guion de datos utilizado para la manipulación y obtención de los datos almacenados en InfluxDB.

Los pasos a seguir para un primer contacto con InfluxDB son la instalación en local, ejecución del servidor, escritura y lectura de datos.

Los pasos de escritura y lectura inicialmente se prueban con la ayuda de una aplicación externa de testeo, después de aprender cómo funciona la comunicación con el servidor, se inicia a probar la escritura y lectura de datos desde LabVIEW y preparar plantillas para su futuro uso en el proyecto.

Con todos los conocimientos necesarios y con la ayuda del equipo del proyecto, se empieza el desarrollo en LabVIEW de las diferentes aplicaciones para la migración.

La primera aplicación es la encargada de modificar y adaptar los archivos TDMS con las etiquetas que se implementan en InfluxDB para el filtrado de los datos.

Con los archivos TDMS adaptados, se procede al desarrollo de la segunda aplicación encargada de convertir los datos almacenados en los archivos TDMS al formato Line Protocol, utilizado por InfluxDB para la escritura de datos.

Llegados a este punto, se considera que la parte de recuperación de los datos históricos está completa a nivel de desarrollos en LabVIEW.

Ahora se empieza con el desarrollo, o más bien, actualización de la aplicación encargada de la lectura de los datos generados por los sensores. En esta aplicación se debe actualizar la funcionalidad de escritura de los datos en archivos TDMS por el nuevo desarrollo para escribir y almacenar los datos en la base de datos InfluxDB.



1.5 Estructura

En esta sección se hace un pequeño recorrido por los capítulos sucesivos del documento detallando superficialmente el contenido de cada uno.

En el **estado del arte** se comenta la situación actual de las tecnologías que se utilizan o intervienen de cierta manera en este desarrollo del proyecto. Después de observar y entender el estado actual de las tecnologías, se hace un estudio de proyectos reales similares al desarrollado, pudiendo hacer comparaciones y entender la situación desde otros puntos de vista.

A continuación, en el capítulo **Análisis del problema**, se analiza el problema que se pretende resolver, se estudian diferentes soluciones posibles, siempre que sea posible.

Para realizar una buena comparación de soluciones, se deben investigar diferentes y nuevas tecnologías que puedan aportar una solución diferente a lo utilizado hasta el momento. Para este capítulo, el estudio realizado en el estado del arte puede ser muy útil.

Con la solución ya elegida, se realiza el **Diseño de la solución**, donde se entra más en detalle con las partes y decisiones técnicas para llevar a cabo el desarrollo. En este capítulo se diseña la arquitectura principal del sistema y se detallan las tecnologías a utilizar.

Llegados a este punto, se puede empezar el **Desarrollo**. En este capítulo se detalla todo el proceso técnico seguido para llevar a cabo los objetivos principales del proyecto.

Con todos los puntos del desarrollo terminados, empieza el capítulo de **Implantación**.

En este capítulo se detalla todo el proceso que se debe seguir para realizar la puesta en marcha de todas las aplicaciones desarrolladas. Esta puesta en marcha va ligada con un capítulo de **Pruebas** que permite corregir los posibles errores que pueden surgir durante el desarrollo.

Por último, se finaliza con una conclusión final de todo el proceso y posibles trabajos futuros o actualizaciones previstas.



2 Estado del arte

Las tecnologías en el mundo de la industria avanzan muy rápido para poder abastecer todas las necesidades, generando que se tenga que reinventar cada día, surgiendo así el termino de Industria 4.0.

Pero en lo que a este proyecto respecta, que es la **Industria 4.0**?

Según un artículo de Deloitte, es la cuarta revolución industrial que está cambiando la forma en que los negocios operan y los entornos en que deberán competir para no perder cuota de mercado, es decir, se busca transformar a la empresa en una organización inteligente con el objetivo de mejorar los resultados de negocio.

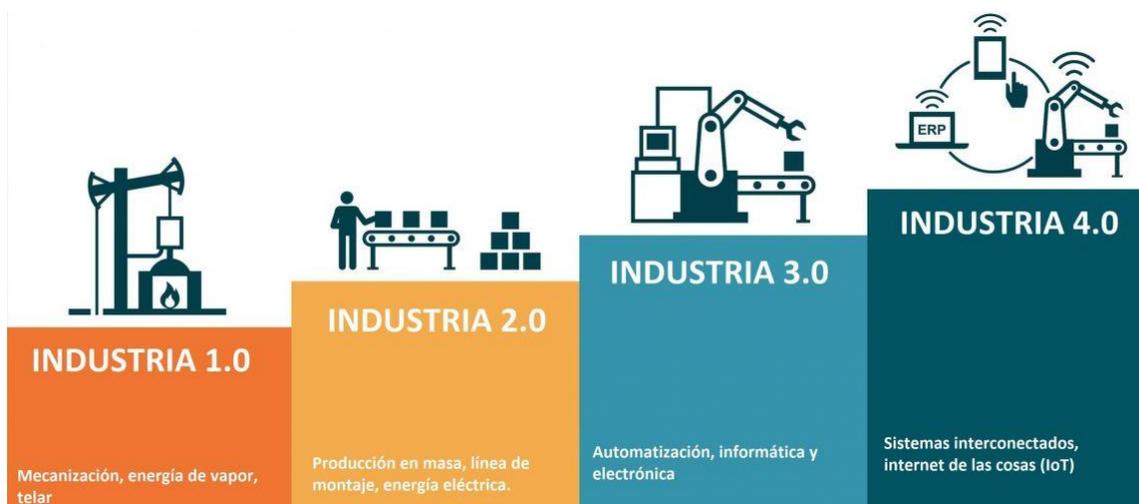


Figura 1 Evolución de las tecnologías de la industria

Este cambio supondrá acceso inmediato a los datos que se generan en los negocios y empresas, junto con la inteligencia de negocio se podrá mejorar e innovar en nuevos desarrollos de productos y servicios. De esta manera se generarán más acciones entre el mundo físico y digital.

El flujo que seguirá la información con los nuevos desarrollos se desglosa en 3 sencillos pasos:

- **Físico a digital:** La información se origina en el mundo físico, se captura y se almacena en un registro digital.
- **Digital a digital:** Aquí se considera el procesamiento e interpretación de los datos y la información mediante inteligencia artificial y análisis.
- **Digital a físico:** Según la información y los datos obtenidos en el paso anterior, realizando una interpretación del resultado se obtienen decisiones y acciones que afectaran en el mundo físico

La importancia de esta nueva revolución industrial no solo afecta a los procesos de fabricación e industria, los cambios son aplicables a cualquier sector de negocio incluida la sociedad, con posibles cambios provocados en la forma en que los clientes interactúen.

Además de los cambios, también se pueden desarrollar productos y servicios nuevos a partir de nuevas implementaciones con los avances tecnológicos como la inteligencia artificial o la robótica que pueden permitir la realización de labores no explotadas hasta la fecha o mejorar la eficiencia y seguridad de otras.

A partir de los sensores y otros sistemas se puede obtener la suficiente información para conocer la energía que se está generando, pero más allá de conocer los datos de producción, también se puede identificar fallos, anomalías o incluso situaciones donde se puede mejorar la producción con pequeños cambios.

Toda esta comunicación necesita de un proceso compatible para todos los dispositivos y es ahí donde surgen muchos problemas pues cada fabricante desarrolla protocolos de comunicación propios que impiden la globalización, pero se está intentando establecer un estándar abierto denominado MQTT (Message Queuing Telemetry Transport) permitiendo que todos los fabricantes participen en el soporte facilitando la comunicación entre dispositivos.

Debido al gran impacto que está habiendo del IoT en la industria, se ha definido el nuevo termino IIoT (Industrial Internet of Things). Cada vez hay más casos que confirman la rentabilidad de inversiones de digitalización y uno de los sectores es las energías renovables.

A continuación, se detalla un caso real, similar al desarrollado en este proyecto, sobre la monitorización de parques eólicos, proyecto muy similar al tratado en este documento.

El siguiente ejemplo ha sido elegido por la similitud tanto en el objetivo del proyecto como en la actualización de sistemas antiguos a nuevos desarrollos con el sistema de almacenamiento InfluxDB.

Vleemo es una empresa desarrolladora de parques eólicos y se enfrenta al reto de monitorización de las condiciones ambientales para el seguro y correcto funcionamiento de las turbinas eólicas. Todos los parámetros de monitorización eran obtenidos por sensores y diferentes sistemas de adquisición de datos implementando tecnologías IoT.

Previamente, las turbinas eran monitorizadas con sistemas SCADA de control remoto. Posteriormente se desarrolló un sistema donde los sensores envían la información a una base de datos InfluxDB.

En InfluxDB se almacena toda la información recogida por el sistema de adquisición de datos que, posteriormente es procesada para mostrar, mediante la aplicación de visualización Grafana, la información más relevante.

Con este ejemplo se demuestra que InfluxDB aporta una de las mejores soluciones a almacenamiento de datos de proyectos IoT donde se generan gran cantidad de datos de diferentes fuentes. Por otra parte, InfluxDB soporta utilidades de tratamiento de datos permitiendo que la parte visual de los datos solo recibirá la información necesaria y relevante para el usuario final.

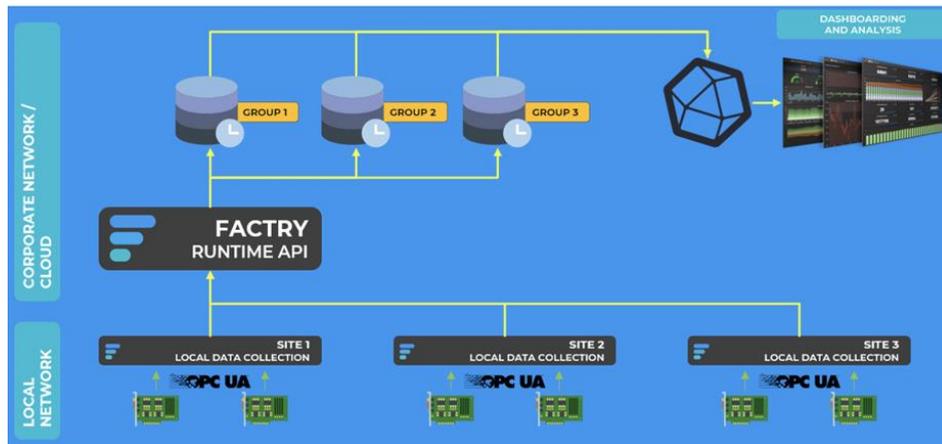


Figura 3 Sistema utilizado por la empresa Vleemo

Tanto el desarrollo de Vleemo como el desarrollo descrito en este documento, generan una cantidad creciente de datos en periodos cortos de tiempo.

En pocas palabras, se ha descrito el Big Data, explicado de forma más detallada es un término que describe el gran volumen de datos, de forma estructurada o sin estructura, que generan desarrollos tecnológicos con implementaciones de sensores y sistemas de adquisición de datos, obteniendo valores del mundo físico o valores digitales a partir de datos ya obtenidos. Pero la parte importante no es la gran cantidad de datos si no el valor de esos datos y los beneficios que pueden aportar a las empresas realizando análisis sobre ellos.

El Big Data se caracteriza con las tres “V”

Volumen: En referencia a la cantidad de datos que se deben procesar, tratándose de datos con o sin estructura.

Velocidad: En referencia al ritmo en que se reciben los datos, esto provoca que se deban transmitir a la memoria en vez de escribirse en un disco duro.

Variiedad: En referencia los diversos tipos de datos disponibles, anteriormente los datos eran estructurados y se podían almacenar en bases de datos relacionales. Ahora con los diversos tipos de contenido para un mismo proyecto se deben implementar soluciones para datos no estructurados.

Con el aumento del uso de los datos, han surgido otras dos características que se podría decir que van relacionadas.

Valor y Veracidad.

Los datos poseen un cierto valor para el desarrollo que los genera, pero dicho valor se puede considerar nulo si los datos no son verídicos.

Tal cantidad de datos, en ocasiones, puede que algunos tipos de software no pueda gestionarlos, InfluxDB ha sido desarrollado pensando en este problema aportando soluciones a almacenamiento masivo de datos.

3 Análisis del problema

Después de la primera versión del producto PSFVIEW, y durante un tiempo de funcionamiento surge el problema de la organización de los datos, por la generación masiva de archivos TDMS con la información monitorizada de los parques fotovoltaicos.

La utilización de archivos TDMS obliga a realizar desarrollos de servicios web en LabVIEW, encargándose de la lectura y procesado de los datos de los ficheros.

3.1 Análisis de soluciones

Las soluciones que se barajan para evitar un colapso de archivos y no generar pérdida de información es centralizar los datos en otro sistema más óptimo para el tipo de producto.

Las soluciones presentadas con sistemas de bases de datos son:

- MariaDB ColumnStore: opción óptima para almacenamiento de datos de series temporales pero construido en el sistema MariaDB Server. Se tiene buen manejo de Maria DB dentro de la empresa, pero no proporciona nuevas funcionalidades.
- InfluxDB: Base de datos desarrollada exclusivamente para series temporales ofreciendo tiempos de acceso bajos y ejecución de gran variedad de funciones para el procesamiento de datos. Se necesita tiempo de aprendizaje sobre la estructura de funcionamiento y el lenguaje de consulta.

3.2 Solución adoptada

Después de analizar ambas soluciones, se decide implementar la migración a la base de datos InfluxDB por las oportunidades que ofrece de procesado y control de los datos.

Debido a que no existe un método implementado para la migración de datos de archivos TDMS a InfluxDB, se desarrollan las aplicaciones necesarias para migrar los datos históricos y también los datos que se obtengan en tiempo real.

4 Diseño de la solución

4.1 Arquitectura del sistema

El diseño de la solución adoptada cuenta con diferentes partes que utilizan diferentes tecnologías.

A continuación, se muestra un diagrama de las partes que compondrán el sistema a desarrollar y se detalla las tecnologías a utilizar en el desarrollo para cada parte.

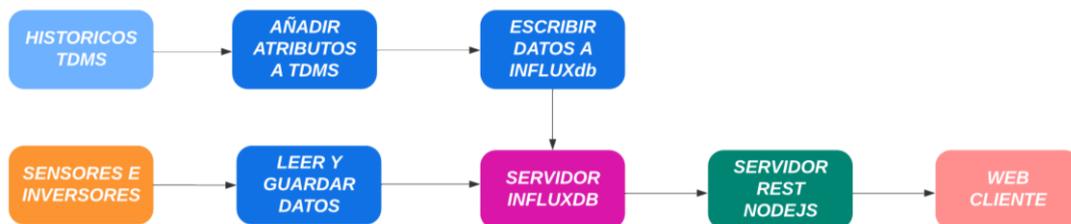


Figura 4 Arquitectura del sistema

Como se puede ver en el diagrama, existen dos ramas que se conectan finalmente en el servidor de InfluxDB, la rama superior es el proceso diseñado para la recuperación de los datos históricos almacenados hasta la fecha de la puesta en marcha.

En el caso de esta primera rama, se puede realizar la implementación asincrónicamente a la puesta en marcha del sistema de monitorización en tiempo real.

En segundo lugar, se tiene el esquema que seguirán los datos que se miden en tiempo real una vez estén todas las aplicaciones desarrolladas y se haga la puesta en marcha del sistema.



Figura 5 Leyenda arquitectura del sistema

En la figura anterior se muestra la asignación de colores a cada tecnología utilizada. En este caso se observa que las tecnologías implicadas en este desarrollo son LabVIEW, InfluxDB, TDMS y Node JS. En el caso de la parte front end, no es relevante en este proyecto pues, las actualizaciones de la parte web se realizan en el servidor REST desarrollado con NodeJS.

4.2 Diseño detallado

La solución adoptada pasa por el desarrollo de 2 aplicaciones y la actualización de otra aplicación, actualmente en funcionamiento.

La primera aplicación es la encargada de modificar y adaptar los archivos TDMS para, posteriormente, que otra aplicación migre los datos. Este proceso se realiza en diferentes aplicaciones pues, el migrado de datos en este caso, es un proceso lento y debe ser seguro para evitar la pérdida o modificación indeseada.

El proceso simplificado de la aplicación consiste en la lectura del archivo TDMS, se añaden las columnas Field, Medida, TagEstacion, TagSistema, TagFamilia con sus datos y tres columnas auxiliares TagAux1-3. Este proceso creará un archivo TDMS nuevo con los anteriores atributos añadidos, de esta forma mantenemos el archivo original intacto para evitar posibles pérdidas de información.

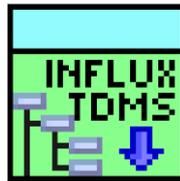


Figura 6 APP_Influx_Set_TDMS_Attributes.vi

El funcionamiento de la aplicación de escritura, dado un directorio donde estarán los archivos que se desean subir a InfluxDB, lee archivo por archivo y de sus datos crea un texto, en formato *Line Protocol* de InfluxDB, donde indica toda la información que se envía vía HTTPS al servidor de InfluxDB que almacena los datos. Si la respuesta HTTP es satisfactoria, se accede al siguiente archivo del directorio. En caso de respuesta de error por parte del servidor InfluxDB, se almacena en la aplicación el archivo del error para comprobar y reintentar la subida posteriormente.

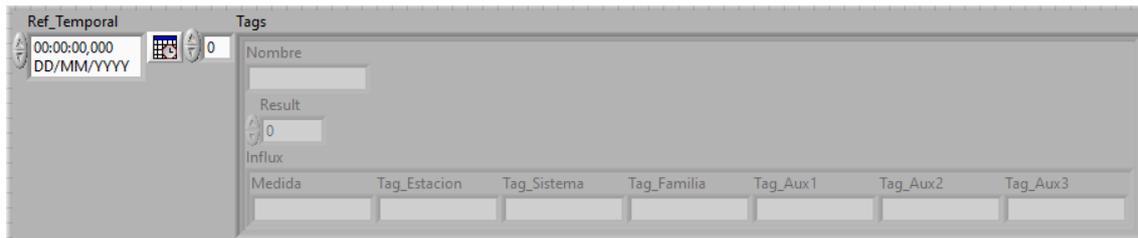


Figura 7 App_TDMS_to_influx_queries.vi

Una vez obtenidos los nuevos archivos TDMS con los nuevos atributos, el siguiente paso es la modificación de las partes de la aplicación que se encargan de almacenar los datos en tiempo real.

Para lograr una modificación más limpia, se desarrollan los subprogramas que dada una entrada de datos se transforman en formato *Line Protocol* de InfluxDB para su inserción.

El formato de los datos de entrada se detalla como un array de clusters con la siguiente estructura:



The image shows a software interface for data entry. On the left, there is a 'Ref_Temporal' section with a date input field showing '00:00:00,000' and 'DD/MM/YYYY' below it. To the right is a 'Tags' section with a 'Nombre' field, a 'Result' field showing '0', and an 'Influx' section. Below these is a table with columns: 'Medida', 'Tag_Estacion', 'Tag_Sistema', 'Tag_Familia', 'Tag_Aux1', 'Tag_Aux2', and 'Tag_Aux3'. Each column has a corresponding empty input field below it.

Figura 8 Formato de entrada de datos

Como se puede observar, desde el primer momento que obtenemos los datos se relacionan con una referencia temporal, necesaria para identificar el instante de cada valor medido.

Con este formato se simula de cierta manera la forma en que se almacenaran dichos datos, lo que facilita la inserción en la base de datos de series temporales.

Cuando los datos pasados al subprograma son procesados, se genera un array de textos con el formato *Line Protocol* para ser enviados e insertados en la base de datos.

4.3 Tecnologías utilizadas

LabVIEW es un entorno de programación gráfica desarrollado por National Instruments, utilizado principalmente por ingenieros para implementar sistemas de automatización, control y monitorización. El lenguaje de programación son bloques gráficos cableados entre ellos para enviar la información de unos a otros. Estos bloques son funciones definidas, condicionales o bucles donde se pueden configurar las diferentes entradas y salidas de datos.

TDMS (Technical Data Management Streaming) desarrollado también por National Instruments, es un formato de almacenamiento de datos con una organización interna jerárquica inherente. Este formato ofrece una cualidad de rapidez junto con sistemas desarrollados en LabVIEW, sin embargo, con la nueva implementación de sistemas web, aparece la posibilidad de accesos simultáneos y exportaciones masivas de datos.

InfluxDB es una base de datos de series temporales que forma parte de un conjunto de aplicaciones desarrolladas por la empresa InfluxData. Este tipo de bases de datos son más rápidas en términos de procesado y almacenamiento de datos de series temporales. Para su utilización se ha implementado el lenguaje de guion de datos Flux, que se ve en los siguientes puntos.

NodeJS es un entorno de tiempo de ejecución en tiempo real de JavaScript permitiendo ejecutar desarrollos con este lenguaje. NodeJS ha sido desarrollado por los creadores de JavaScript. En nuestro caso, NodeJS permite desarrollar una aplicación que permanezca siempre a la escucha en un puerto determinado y que realice acciones en función del tipo de entradas y salidas que generen ciertos eventos. En pocas palabras, permite desarrollar un servidor REST.

5 Desarrollo

En este capítulo, se desarrollan todos los pasos pensados y diseñados para llevar a cabo y conseguir los objetivos marcados.

5.1 Aprendizaje de las tecnologías

5.1.1 LabVIEW

LabVIEW es un entorno de programación gráfica que utiliza lenguaje de bloques gráficos cableados. Este software de programación es propiedad de la empresa National Instruments. Desarrollado en la década de los 80, LabVIEW continúa siendo uno de sus productos más destacados actualmente con la última versión 2021 SP1 lanzada el 16 de febrero de 2022

EL entorno gráfico de LabVIEW cuenta con tres tipos de ventanas a la hora de abrir y manipular un proyecto.

Inicialmente dispone de la ventana **Project Explorer** donde se muestra de forma esquematizada la estructura interna de directorios y archivos con los que cuenta cada proyecto, facilitando la navegación por este.

Esta estructura parte de la raíz del proyecto y representa el equipo local donde se encuentra el proyecto.

A continuación, se detalla un tipo de estructura organizativa para proyectos simples:

- **Controls:** ubicación de los archivos de controles definidos y utilizados en el proyecto.
- **SubVI:** ubicación de los archivos de funciones definidos y utilizados en el proyecto.
- **Archivo Principal:** Archivo con extensión .vi donde se implementa el programa principal del proyecto, es decir, donde se utilizan las funciones y controles que dan forma a la funcionalidad del programa principal.
- **Dependencies:** Ubicación de todos los archivos de librerías instaladas y utilizadas en el desarrollo de la aplicación.
- **Build Specifications:** Incluye configuraciones de compilación de distribuciones fuente y otros tipos de compilaciones.

A continuaci3n, se muestra una figura de la ventana **Project Explorer** de un proyecto de ejemplo donde se pueden observar los diferentes directorios mencionados.

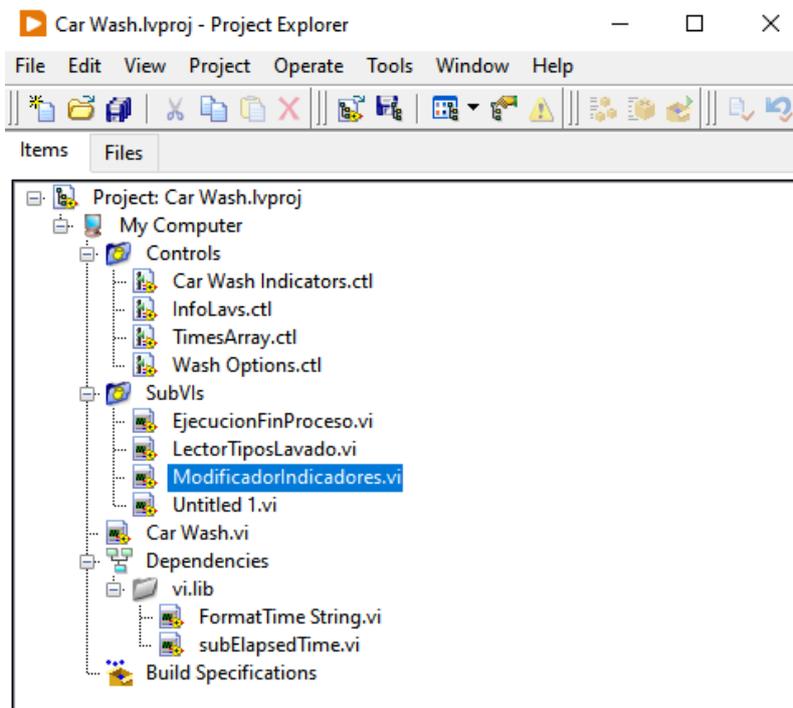


Figura 9 Ejemplo de la ventana Project Explorer

En la imagen anterior se pueden observar los diferentes tipos de archivos de los que consta un proyecto desarrollado en LabVIEW, los cuales se describen a continuaci3n.

- **.lvproj**: Archivo que contiene toda la informaci3n y estructura del proyecto.
- **.vi**: Archivo con la implementaci3n l3gica que se desarrolla en el proyecto, cuenta con dos tipos de ventanas: Front Panel y Block Diagram.
- **.ctl**: Objeto del Panel frontal definido para su uso en diferentes archivos VI.

Cuando se crea o se modifica un archivo VI se muestran las ventanas Front Panel y Block Diagram.

En la ventana Front Panel se diseña la disposici3n de los controles e indicadores de los que constara el archivo VI. Estos controles e indicadores permitirán al usuario poder introducir, modificar y visualizar los valores de las variables que se manipulen en el programa.

En el caso de los archivos SubVI, estos Controles e Indicadores serán las entradas y salidas respectivamente de la funci3n que desempeñe. Estas entradas y salidas se pueden configurar en el Panel de conexi3n que se puede ver en la figura siguiente.

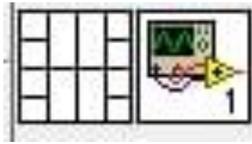


Figura 10 Ejemplo de Panel de Conexión

Por convenio, los cuadrados de la parte izquierda se utilizan como datos de entrada y los de la parte derecha como datos de salida, su configuración se realiza pulsando en el cuadrado deseado y posteriormente en el control o indicador que se desee utilizar como entrada o salida.



Figura 11 Ejemplo de Front Panel

En la imagen anterior se puede observar un ejemplo de Front Panel, donde se tienen botones y barras selectoras que generan datos de entrada al programa y por otra parte indicadores luminosos y numéricos que muestran los datos del programa.

La última ventana grafica del entorno LabVIEW es Block Diagram, la ventana donde se desarrolla la parte lógica del programa.

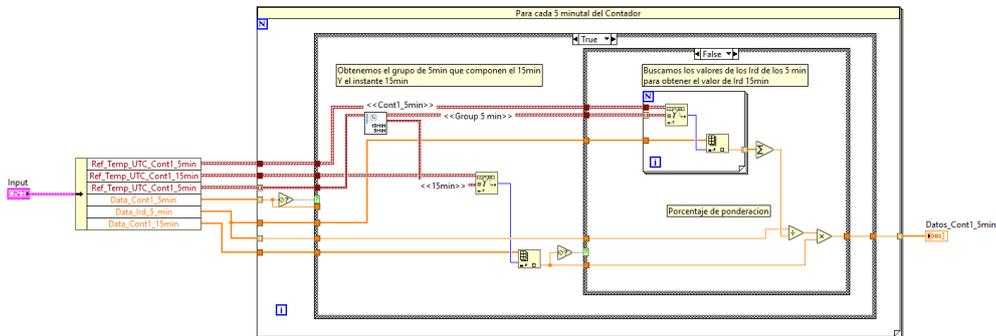


Figura 12 Ejemplo de código en LabVIEW

En el ejemplo de la imagen anterior se pueden observar diferentes funcionalidades y estructuras como un bucle for, condiciones y funciones definidas en SubVIs.

Siguiendo el estándar, la lectura y secuencia de ejecución del programa se realiza de izquierda a derecha. La organización del código es fundamental para facilitar su lectura en posibles modificaciones.

Todas las funcionalidades, controles e indicadores se pueden añadir desde la paleta de funciones, ahí se encuentran todas las funciones definidas en LabVIEW y las añadidas posteriormente en librerías. De esta manera, solo se debe buscar la funcionalidad deseada, ubicarla en el código y conectar las diferentes entradas y salidas.

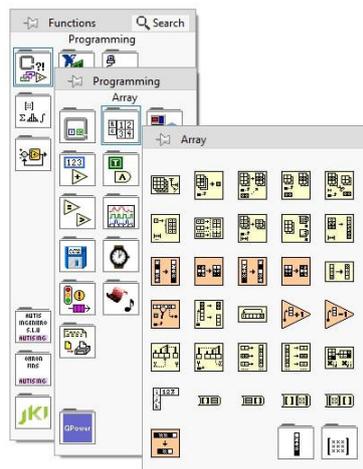


Figura 13 Paleta de Toolkits y librerías

En el caso de querer añadir un SubVI desarrollado por uno mismo, será suficiente con buscarlos en el explorador del proyecto y arrastrarlo hasta la ventana de bloques.

5.1.2 TDMS

En múltiples ocasiones se puede encontrar información almacenada donde la descripción de dicha información es escasa o inexistente. En estas situaciones se hace realmente complicado localizar o identificar un particular conjunto de datos.

La empresa National Instruments se dio cuenta que se necesitaba un formato de almacenamiento capaz de remediar estas deficiencias y por tanto desarrolló el formato TDMS (Technical Data Management Streaming).

El formato TDMS se desarrolló de forma propia y ajustada a las demás aplicaciones de NI, permitiendo unas altas velocidades de lectura y escritura.

A continuación, se detalla la estructura de los archivos TDMS:

La estructura interna se basa en tres niveles jerárquicos que son Archivo, Grupo y Canal.

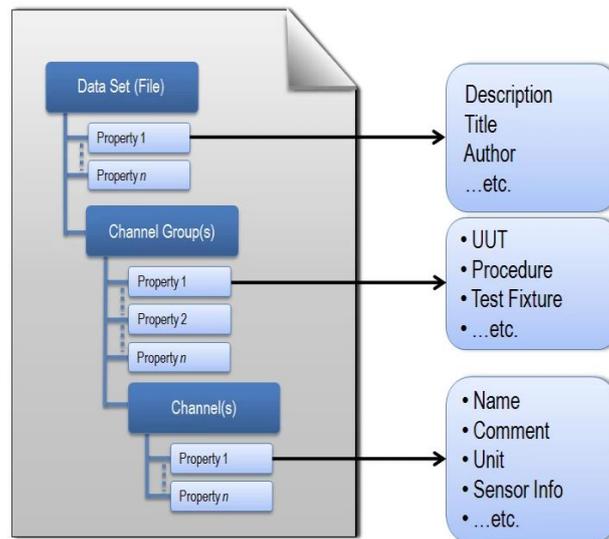


Figura 14 Estructura interna de ficheros TDMS

Como se observa en la imagen, cada nivel jerárquico puede contener un número ilimitado de objetos del siguiente nivel, es decir, el nivel archivo puede contener ilimitados grupos y estos pueden contener ilimitados canales.

Por otra parte, en cada nivel además se pueden almacenar ilimitadas propiedades y a cada una de estas propiedades se le pueden definir infinidad de atributos.

Esto muestra el alto nivel de organización y documentación de la información almacenada que se traduce en la rapidez de lectura y escritura que dispone este tipo de archivos.

Por otra parte, evita el hecho del rediseño debido a la facilidad de amplificar el sistema adoptado a los nuevos formatos o nuevos datos.

5.1.3 InfluxDB

InfluxDB es una base de datos de series temporales, destinada al almacenamiento de alta velocidad y disponibilidad de datos referenciados por marcas de tiempo. Opción que interesa bastante para proyectos de IoT o Big Data, permitiendo recopilar, almacenar, monitorear, visualizar y alertar en función de los datos y configuraciones.

InfluxDB ha sido desarrollado desde cero por la empresa InfluxData y lanzada por primera vez en el año 2013.

Puesto que InfluxDB fue una apuesta de creación desde cero, se implementó inicialmente un lenguaje de consulta llamado InfluxQL, similar a SQL, pero no idéntico, pues una base de datos relacional puede manejar series temporales, pero no está optimizada para ello, sin embargo, InfluxDB puede almacenar grandes cantidades de series temporales y realizar procesamiento con gran eficacia.

A partir de la versión de InfluxDB 1.8.0 se implementó el lenguaje de scripting Flux para mejorar características y añadir otras nuevas.

InfluxDB cambia la forma de ver y entender una base de datos tradicional, eliminando el concepto de tabla como se conoce en otras implementaciones.

En InfluxDB se tienen buckets como forma de almacenamiento de los datos temporales, todos los buckets disponen de un periodo de retención de la información pudiendo ser infinito.

Al ser temporal, el parámetro principal de cualquier consulta que se desee realizar será un rango temporal. Esto implica que toda información almacenada en esta base de datos deberá disponer de un valor temporal, en caso de no haber indicado en el momento del almacenamiento de dicho dato, automáticamente se le asignará el instante en el cual el dato es almacenado.

Los datos que se almacenan, además de contener un instante de tiempo asociado, disponen de etiquetas definidas para una mejor identificación y filtrado. De forma análoga a las bases de datos relacionales, las etiquetas son como los nombres de las columnas de las tablas. Pero no todas las etiquetas son iguales, existen las etiquetas “_measurement” y “_field” que no permiten modificar su nombre, solo permiten asignarle un valor en cada registro.

Las etiquetas personalizables por el desarrollador son grupos de pares clave-valor por punto que permiten realizar relaciones entre datos y a la vez organizar los datos de forma eficiente ante búsquedas.

El sistema que desarrolla InfluxDB no solo se queda en almacenamiento y visualización de los datos almacenados.

InfluxDB dispone de más funcionalidades como la configuración de Dashboards, la creación de tareas programadas o la programación de alertas ante diferentes situaciones de los datos almacenados.

Los paneles de visualización de datos permiten monitorizar las medidas en una variedad de gráficos, de esta forma en una misma ventana se pueden observar diferentes datos y permite realizar comparaciones facilitando a los usuarios la monitorización de los datos.

Las tareas programadas son de gran ayuda para el control de los datos de entrada o los ya almacenados. Esta funcionalidad permite ejecutar un script de Flux de forma periódica y programada, de esta manera se pueden realizar tareas de control las 24 horas del día.

Por último, otra funcionalidad imprescindible cuando se trata de monitorización de datos son las alertas. InfluxDB permite configurar alertas enviar avisos mediante diferentes métodos en el caso que se dejen de recibir datos o que estos no coincidan con los umbrales especificados.

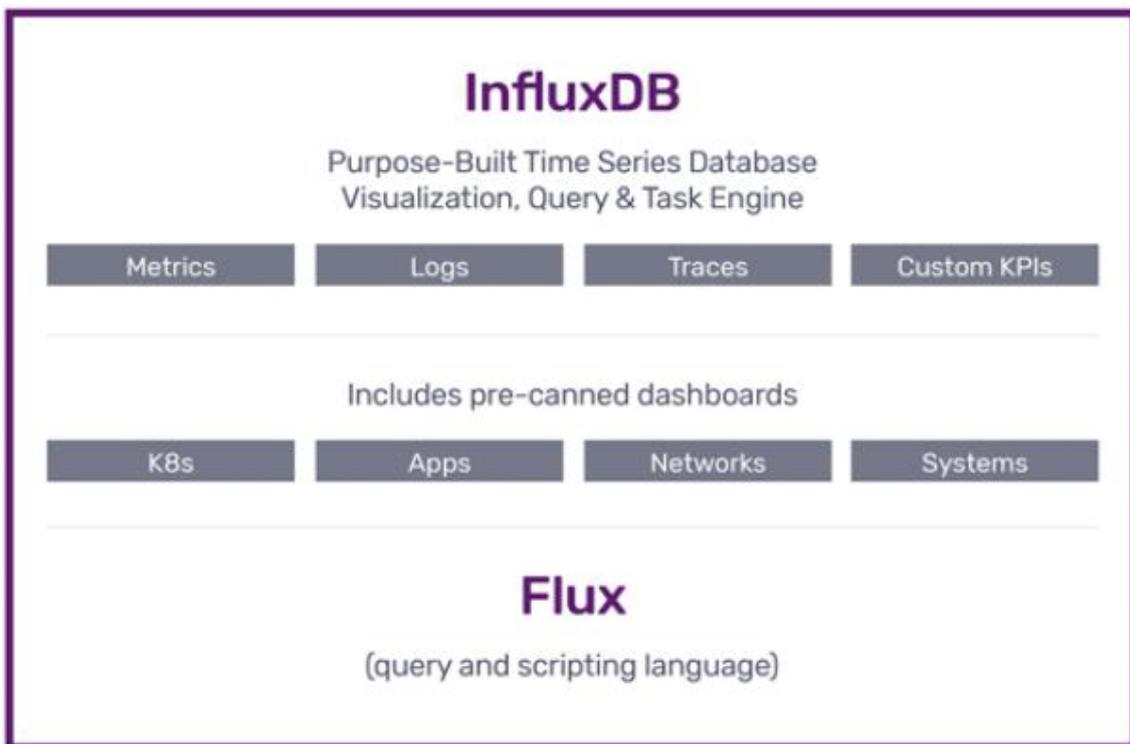


Figura 15 Funcionalidades y características de InfluxDB

5.2 Instalación y puesta en marcha de InfluxDB

5.2.1 Instalación del servicio de InfluxDB

En este punto se aborda la instalación del servidor para la base de datos InfluxDB y InfluxCLI, un cliente para realizar operaciones de interacción y configuración sobre la base de datos.

Para la instalación de ambos, se descargan los paquetes de instalación por separado de la web oficial de InfluxData.

En los paquetes descargados se encuentran los dos archivos ejecutables para su ejecución y en el caso de InfluxDB. La instalación del servidor InfluxDB se realiza como servicio de Windows, de esta forma no será necesario su inicio manual en el arranque del sistema. Para la instalación del servidor de InfluxDB como servicio de Windows se cuenta con un software de terceros que permite crear servicios de Windows sobre archivos ejecutables, el software es NSSM, muy sencillo de utilizar.

Se sigue el estándar de nomenclatura de directorios utilizado en la documentación oficial, por tanto, el primer paso en la instalación es crear los directorios donde se guardan los archivos ejecutables `influxd.exe` y `influx.exe`, servidor de InfluxDB y cliente respectivamente.

Los directorios creados son:

- *C:\Program Files\InfluxData\Influxdb (Servidor de InfluxDB)*
- *C:\Program Files\InfluxData\Influx (Cliente InfluxDB)*

Con los directorios creados, se pueden descomprimir los paquetes descargados en sus respectivos directorios, asegurándose que los archivos ejecutables se encuentran en los nuevos directorios.

Como se ha comentado anteriormente, para la instalación del servidor InfluxDB como servicio, se necesita un software de terceros. Se guarda el archivo `nssm.exe` en el directorio donde se encuentra el ejecutable `influxd.exe`.

Con todos los archivos es sus directorios se puede empezar el proceso de instalación, para ello se abre una ventana de línea de comandos de Windows en modo Administrador y se accede hasta el directorio donde se encuentra el software NSSM y el servidor InfluxDB.

```
> cd C:\Program Files\InfluxData\Influxdb
```

En la ubicación del servidor de InfluxDB se ejecuta el siguiente comando para iniciar el software NSSM y poder configurar el servicio de Windows a crear.

> *nssm install influxdb*

En la interfaz del software NSSM se selecciona la ruta absoluta del archivo a convertir en servicio, se especifica el nombre del servicio y se instala.

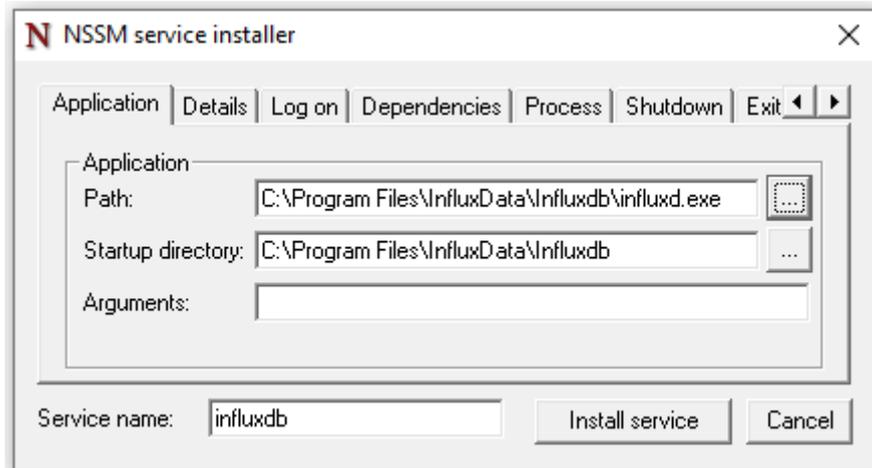


Figura 16 Creación del servicio de InfluxDB

Una vez el servicio instalado y por consola, se ejecuta el siguiente comando que inicia el funcionamiento del servidor InfluxDB como servicio de Windows.

> *nssm start influxdb*

Una vez se realiza el proceso de instalación y se arranca el servicio, se puede comprobar el funcionamiento en el Administrador de tareas, en la sección de servicios. Si el servicio se encuentra en ejecución se puede confirmar que la instalación ha sido exitosa.

Cabe destacar que la primera vez que se inicia la base de datos, se crea la carpeta `.influxbv2` en el directorio del usuario que ha instalado el servidor. En nuestro caso se ha instalado como administrador, por tanto, se encuentra en el directorio del usuario administrador del sistema Windows en el que estemos.

Este nuevo directorio almacena toda la información de la base de datos, incluidos los usuarios, organizaciones y demás configuraciones que se pueden realizar desde la web que se explica en el siguiente punto.

5.2.2 Acceso a la web cliente

En este punto del proceso se tiene un servidor InfluxDB corriendo en el dispositivo Windows donde se ha instalado, ahora es momento de acceder a la interfaz que proporciona para configurar todo lo necesario para empezar a almacenar información.

Para ello, desde un navegador, acceder a la url `http://localhost:8086` y se muestra una web de bienvenida donde se crea un primer usuario, el cual dispone de todos los privilegios, se indica el nombre de la organización y se permite la opción de crear un bucket inicial.

Una organización es un espacio de trabajo para un grupo de usuarios. Se puede crear más de una organización en una implementación de InfluxDB local, pero no está disponible el uso de más de una organización en el servicio en la nube de InfluxDB que ofrece InfluxData.

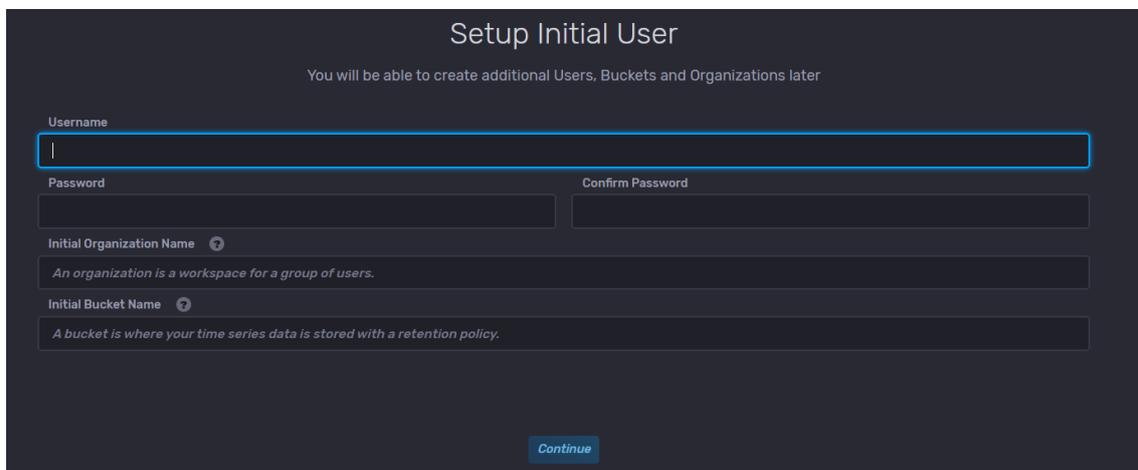


Figura 17 Configuración de usuario y organización

Una vez dentro de la web del servidor InfluxDB y antes de entrar en detalles de configuración para el almacenamiento de datos, se destacan dos partes importantes para la utilización y cambios de configuración.

La primera sección a destacar es acerca de la organización, en esta sección se permite ver los ID's de tanto del usuario que ha iniciado la sesión como de la organización en la que se está trabajando. Estos ID's se deben indicar en ciertas acciones de configuración. También se permite modificar el nombre de la organización, aunque no es una práctica recomendable pues puede causar fallos en el funcionamiento de implementaciones en producción.

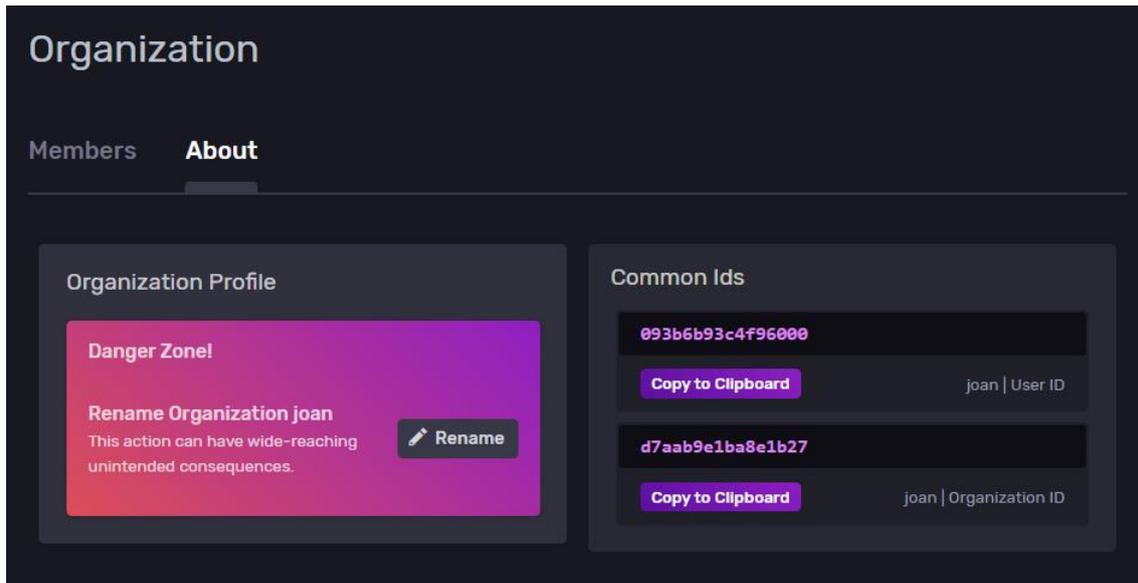


Figura 18 Vista sobre la información de la organización

La segunda sección por destacar es donde se listan los Tokens que dan acceso a las funcionalidades del servidor InfluxDB. Estos Tokens se pueden generar otorgando los permisos que se desee para cada tipo de interacción con el servidor. Por defecto se crea un Token con todos los permisos para el usuario inicial.

Seleccionando el token deseado se muestra una ventana donde se indican los permisos que dispone.

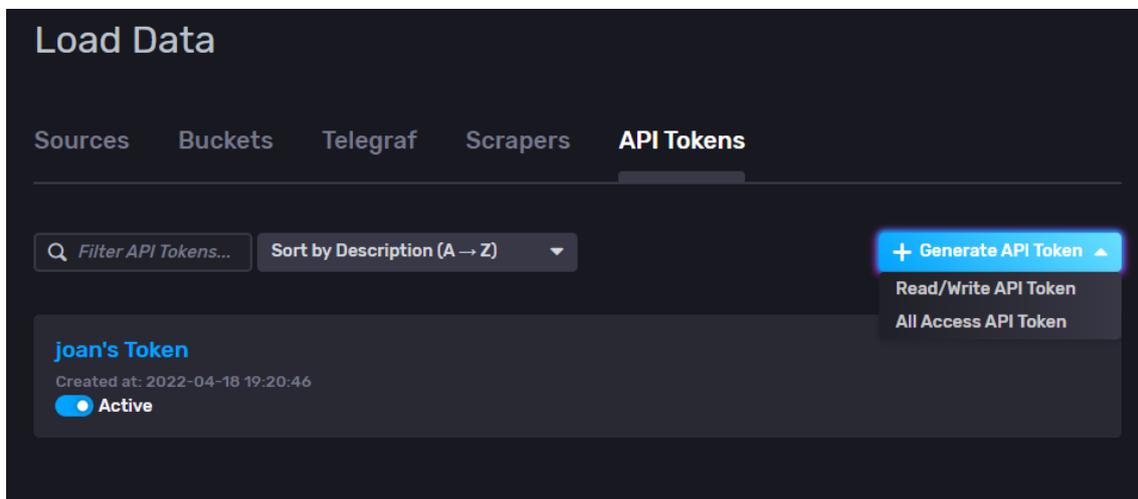


Figura 19 Sección para obtener el token de acceso

5.3 Pruebas de funcionamiento

5.3.1 Escritura de datos

Después de terminar todo el proceso de instalación de la base de datos InfluxDB se realizan unas pruebas para el aprendizaje del funcionamiento de las acciones de escritura y lectura de los datos. Estas acciones se pueden realizar de más de una forma, inicialmente se realizan pruebas con la web cliente y con un programa de terceros llamado Postman, programa de testeo de API's y depuración de aplicaciones web.

Antes de empezar con la escritura y lectura de datos, es necesario crear un bucket donde se almacenen todos los registros que se desean escribir en la base de datos. Para ello, utilizando la web cliente de InfluxDB, en la pestaña Load Data > Buckets se selecciona Create Bucket. Se debe indicar el nombre del Bucket y seleccionar el tiempo que permanecerán los datos almacenados en la base de datos antes de ser eliminados, en este ejemplo se selecciona la opción de 30 días.

Figura 20 Crear un bucket de almacenamiento

Una vez creado el bucket se inicia con la escritura de datos utilizando el protocolo propio de formato de los datos para ser insertados llamado Line Protocol que consta de la siguiente estructura:

```
measurementName,tagKey=tagValue fieldKey="fieldValue" 1465839830100400200
-----
|           |           |           |
Measurement Tag set      Field set  Timestamp
```

Figura 21 Formato de escritura de datos

En el conjunto de Tags, se pueden insertar más de uno de igual forma que el conjunto de Fields pero este último no es recomendado en algunas circunstancias donde se pretende filtrar solo un tipo de valor por cada conjunto de Tags.

En los próximos ejemplos se insertan datos simulando el caso de la monitorización de la temperatura y humedad de una casa con dos plantas y 3 habitaciones por planta. A continuación, se muestra una sentencia de muestra de Line Protocol.

> Residencia,planta="Baja",habitacion="cocina" temperatura=25,humedad=65 1652725800

La escritura de datos desde la web cliente se puede realizar de dos formas con el protocolo:

- Copiar las líneas de cada registro, de forma manual, en el bloque de entrada de texto. Forma recomendada para pocas líneas o depurar el formato.
- Cargar un fichero de texto que contenga los registros en el formato Line Protocol. Forma recomendada para subir archivos con grandes cantidades de líneas.

Dato importante a tener en cuenta, el carácter que se añade al final de cada línea, pues InfluxDB utiliza “\n” para identificar y poder separar un registro del siguiente. Es posible obtener un error al intentar subir datos desde un fichero debido a que el editor con el que se estén escribiendo los registros añade un carácter diferente al final de cada línea.

Después de insertar las medidas de 4 minutos consecutivos, se realiza la prueba de insertar más datos consecutivos desde el programa de terceros Postman para testear la escritura de datos por peticiones HTTP. Para ello, se consulta en la documentación oficial, los métodos de la API de escritura de datos. Se observa que la escritura se realiza con una petición POST a la URL con la siguiente ruta:

`/api/v2/write? bucket=ClimaInterno&org=Joan&precision=s`

5.3.2 Consulta de datos

Después de crear y añadir datos a un bucket, el siguiente paso importante es recuperar los datos que se han almacenado para mostrarlos en la aplicación o web del cliente. Antes de pasar a obtener los datos para la aplicación que aún no está implementada, se realizan pruebas de verificación y lectura de los datos escritos anteriormente.

Para ello, inicialmente se realizan pruebas sobre la web de configuración que nos ofrece el servidor de InfluxDB. Por tanto, se accede a la opción Data Explorer del menú lateral, en la parte inferior se muestra el apartado Query Builder donde se puede crear una consulta de los datos en relación a las opciones que se elija, de esta manera se pueden hacer consultas sin conocimientos de Flux.

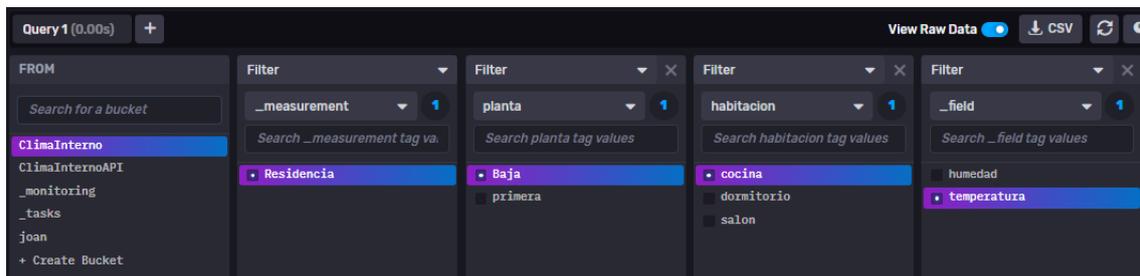


Figura 22 Consulta con el constructor de consultas

También existe la opción de implementar consulta en Flux, incluso ver la consulta que se ha realizado con la selección, pulsando en el botón Script Editor

```

Query 1 (0.04s) +
1 from(bucket: "ClimaInterno")
2   |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
3   |> filter(fn: (r) => r["_measurement"] == "Residencia")
4   |> filter(fn: (r) => r["planta"] == "Baja")
5   |> filter(fn: (r) => r["habitacion"] == "dormitorio")
6   |> filter(fn: (r) => r["_field"] == "temperatura")
7   |> yield(name: "mean")
  
```

Figura 23 Consulta con Flux

Estas consultas generan una respuesta del servidor InfluxDB con los datos correspondientes y estos datos se representan en la web cliente en forma de grafico o forma de tabla para una interpretación más detallada. Esto se puede ver en la opción Data Explorer del menú vertical.



Figura 24 Visualización de datos en gráficos

table	_measurement	_field	_value	_start	_stop	_time	habitacion	planta
MEAN	GROUP STRING	GROUP STRING	NO GROUP DOUBLE	GROUP DATETIME: RFC3339	GROUP DATETIME: RFC3339	NO GROUP DATETIME: RFC3339	GROUP STRING	GROUP STRING
0	Residencia	humedad	65	2022-05-15T21:09:21.838Z	2022-05-17T21:09:21.838Z	2022-05-16T18:30:00.000Z	cocina	Baja
0	Residencia	humedad	65	2022-05-15T21:09:21.838Z	2022-05-17T21:09:21.838Z	2022-05-16T18:31:00.000Z	cocina	Baja
0	Residencia	humedad	75	2022-05-15T21:09:21.838Z	2022-05-17T21:09:21.838Z	2022-05-16T18:32:00.000Z	cocina	Baja
0	Residencia	humedad	55	2022-05-15T21:09:21.838Z	2022-05-17T21:09:21.838Z	2022-05-16T18:33:00.000Z	cocina	Baja
0	Residencia	humedad	65	2022-05-15T21:09:21.838Z	2022-05-17T21:09:21.838Z	2022-05-16T21:16:40.000Z	cocina	Baja
0	Residencia	humedad	75	2022-05-15T21:09:21.838Z	2022-05-17T21:09:21.838Z	2022-05-16T21:17:40.000Z	cocina	Baja
0	Residencia	humedad	85	2022-05-15T21:09:21.838Z	2022-05-17T21:09:21.838Z	2022-05-16T21:18:40.000Z	cocina	Baja
0	Residencia	humedad	65	2022-05-15T21:09:21.838Z	2022-05-17T21:09:21.838Z	2022-05-16T21:19:40.000Z	cocina	Baja

Figura 25 Visualización de datos en tabla

Hasta el momento, los datos no han salido del sistema InfluxDB y para este proyecto se requiere que InfluxDB responda a consultas externar realizadas por otras aplicaciones de terceros. Para ello, utilizando de nuevo el programa Postman, se realizan consultas y se estudia el formato en el cual InfluxDB devuelve los datos almacenados.

De la API se tiene que la URL para la consulta, con método POST es:

<http://localhost:8086/api/v2/query?org=Joan>

En las siguientes imágenes se muestra la petición realizada con todos los parámetros y datos necesarios, según lo consultado en la API de InfluxDB.

```
1 curl --location --request POST 'http://localhost:8086/api/v2/query?org=Joan' \  
2 --header 'Content-Type: application/vnd.flux' \  
3 --header 'Authorization: Bearer  
   sHQrgrP4yLeIbJgcVgt3mfMZkKQCX3dh38mOCC702FKnfJGf_kvNwrCtRWmpipazFBn7g7uI0XPHUDU  
   QA-auWg==' \  
4 --data-raw 'from(bucket: "ClimaInterno")  
5 |> range(start: 2022-05-15T21:09:21.838Z, stop: 2022-05-17T21:09:21.838Z)  
6 |> filter(fn: (r) => r["_measurement"] == "Residencia")  
7 |> filter(fn: (r) => r["planta"] == "Baja")  
8 |> filter(fn: (r) => r["habitacion"] == "cocina")  
9 |> filter(fn: (r) => r["_field"] == "humedad")  
10 |> yield(name: "mean")'
```

Figura 26 Consulta con Flux desde Postman

Con una respuesta exitosa del servidor, se reciben los datos con el formato predeterminado.

```
1 ,result,table,_start,_stop,_time,_value,_field,_measurement,habitacion,planta  
2 ,mean,0,2022-05-15T21:09:21.838Z,2022-05-17T21:09:21.838Z,2022-05-16T18:30:00Z,65,humedad,Residencia,cocina,Baja  
3 ,mean,0,2022-05-15T21:09:21.838Z,2022-05-17T21:09:21.838Z,2022-05-16T18:31:00Z,65,humedad,Residencia,cocina,Baja  
4 ,mean,0,2022-05-15T21:09:21.838Z,2022-05-17T21:09:21.838Z,2022-05-16T18:32:00Z,75,humedad,Residencia,cocina,Baja  
5 ,mean,0,2022-05-15T21:09:21.838Z,2022-05-17T21:09:21.838Z,2022-05-16T18:33:00Z,55,humedad,Residencia,cocina,Baja  
6 ,mean,0,2022-05-15T21:09:21.838Z,2022-05-17T21:09:21.838Z,2022-05-16T21:16:40Z,65,humedad,Residencia,cocina,Baja  
7 ,mean,0,2022-05-15T21:09:21.838Z,2022-05-17T21:09:21.838Z,2022-05-16T21:17:40Z,75,humedad,Residencia,cocina,Baja  
8 ,mean,0,2022-05-15T21:09:21.838Z,2022-05-17T21:09:21.838Z,2022-05-16T21:18:40Z,85,humedad,Residencia,cocina,Baja  
9 ,mean,0,2022-05-15T21:09:21.838Z,2022-05-17T21:09:21.838Z,2022-05-16T21:19:40Z,65,humedad,Residencia,cocina,Baja
```

Figura 27 Respuesta desde InfluxDB

Como se puede ver en la imagen superior, la primera fila de información devuelta en la consulta es el nombre de cada dato, por ejemplo, los Tags o las referencias temporales, las siguientes filas son los registros que han sido devueltos respecto a la consulta realizada.

5.3.3 Desarrollo de la API de pruebas y aprendizaje

Después de conocer y realizar pruebas con InfluxDB a nivel de escritura y consultas de datos, es hora de realizar comunicaciones con la base de datos desde un desarrollo que imite la aplicación final. Por este motivo se decide implementar una aplicación básica a nivel de programación en LabVIEW que pueda comunicar con la base de datos y con vista de futura para ser utilizada como herramienta de aprendizaje a los integrantes del proyecto que requieran una formación para utilizar InfluxDB.

El nombre de la aplicación es Influx_API, permitiendo que el usuario, de forma interactiva, realice acciones en la base de datos y pueda ver el código que permite realizar esas acciones para que pueda ser replicado en otras implementaciones.

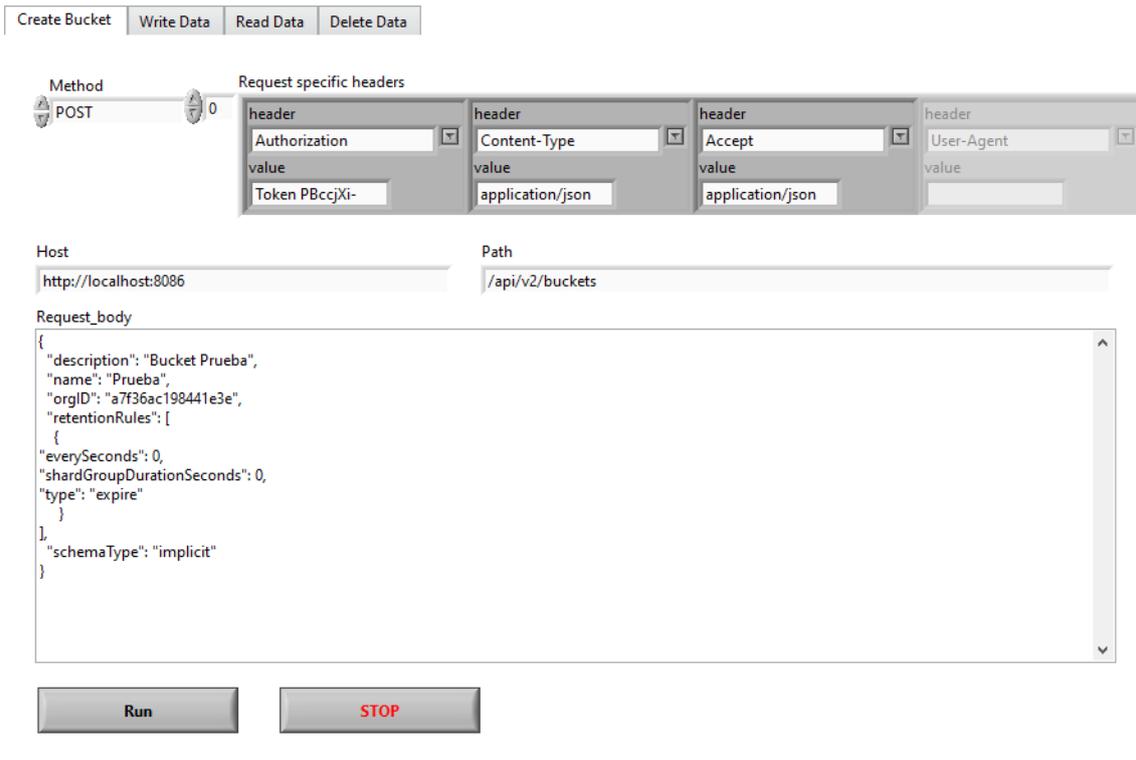


Figura 28 Ventana de la aplicación de LabVIEW para InfluxDB

Inicialmente se implementan cuatro funcionalidades consideradas básicas para iniciarse. De esta forma se puede realizar todo el proceso que se pretende llevar a cabo en la aplicación final, es decir, desde la creación de un bucket, escritura de datos, consulta de esos datos y la opción de eliminar los datos en el caso que se desee.

Por defecto, la aplicación dispone de ejemplos ya implementados, tanto en la sección de Request Body y Path como algunos parámetros de Request specific header. Estos ejemplos o parámetros se ajustan a la acción seleccionada.

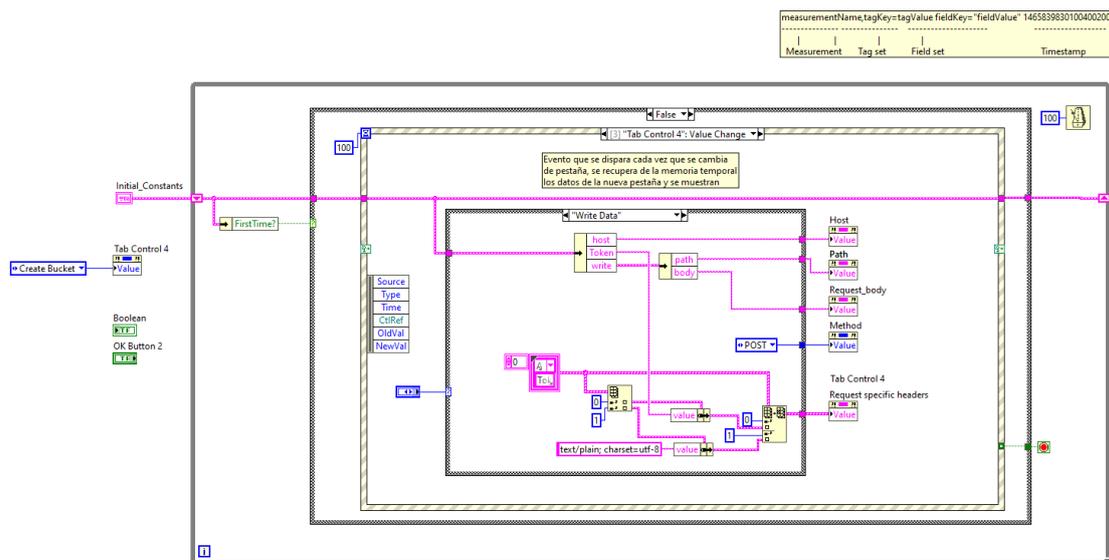


Figura 29 Ejemplo de código de la aplicación

A grandes rasgos, la aplicación cuenta con cuatro procesos que describen todo el funcionamiento interno. La diferencia entre los procesos y las funcionalidades que muestra el menú son, los procesos son lanzados cuando se captura un evento o acción en la aplicación. Estos procesos son: Inicialización de variables, cambiar los datos en función de la pestaña seleccionada, ejecutar la funcionalidad que este seleccionada y detener la aplicación.

Inicialización: Se encarga de inicializar las variables y los campos de entrada de datos con los datos por defecto que se almacenan en la aplicación.

Cambio de pestaña: Identifica que opción del menú se ha elegido y carga los datos correspondientes en los campos de entrada de datos y las diferentes variables.

Ejecutar: Recopila todos los datos de los campos de entrada, identifica los datos y la opción seleccionada y ejecuta el código para realizar la funcionalidad elegida.

Detener: Detiene la aplicación en ejecución.

Cuando ocurre un cambio de pestaña, se realiza una lectura de todos los campos de entrada de datos y se almacena la información en una variable de memoria temporal que persistirá hasta que se modifique otra vez desde el campo de entrada o se cierre la aplicación. De esta forma, permite poder cambiar entre pestañas sin perder los cambios que se realicen.

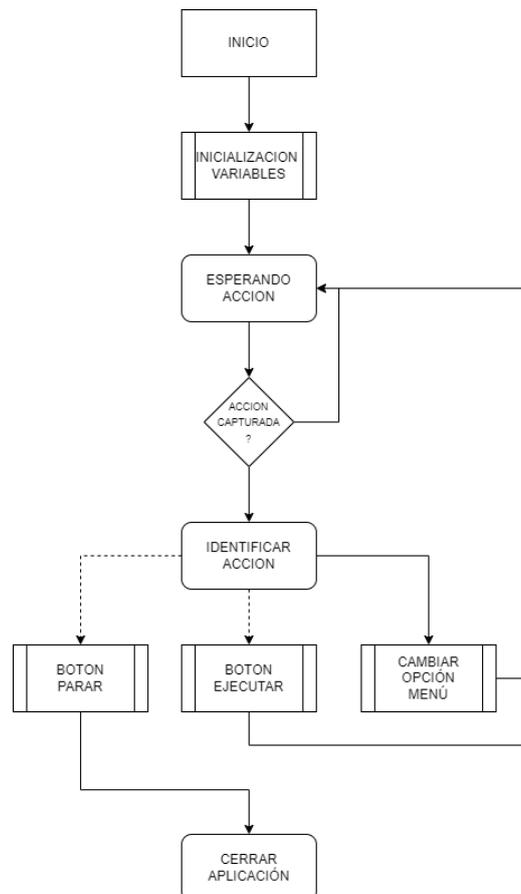


Figura 30 Diagrama de flujo de la aplicación

5.4 Desarrollo de aplicaciones de migración

5.4.1 Aplicación de atributos

Llegados a este punto, se tienen los suficientes conocimientos sobre LabVIEW e InfluxDB y es momento de añadir datos históricos, pero reales, almacenados en archivos TDMS.

Como se describe en el apartado de Objetivos, los archivos TDMS deben ser modificados añadiendo los Tags como propiedades en formato clave – valor, que se utilizan para el almacenamiento y filtrado en InfluxDB.

Para automatizar este proceso en todos los archivos históricos de datos, se implementa la aplicación App_Influx_set_TDMS_Atributos.

El funcionamiento de esta aplicación inicia abriendo un archivo TDMS de un directorio previamente indicado por el usuario, una vez abierto el archivo, se obtienen una lista de los grupos y canales que tiene almacenados. En primer lugar, se obtiene el listado de grupos y por cada grupo se obtiene el listado de canales.

Paralelamente, se realiza una consulta a una base de datos relacional, como es MariaDB, donde se almacenan los diferentes valores que pueden interpretar cada Tag que se asigne a los valores. De esta forma, los valores de los Tags pueden ser modificados o aumentados sin necesidad de modificar la aplicación.

Obtenidos los valores de los Tags para cada canal, se añaden al nuevo archivo TDMS como nombre de propiedad y valor de propiedad.

Tags 10.0_INVERSOR_1_1									
Channel	Datatype	Unit	Length	Minimum	Maximum	Description	NL_ArrayColu	Start Index	
Ref_Tempora	DT_DATE		8640			Ref_Temporal		0	
1100000	DT_DOUBLE		8640			STATUS		0	
1100009	DT_DOUBLE		8640			POTENCIA APARENTE		1	
1100010	DT_DOUBLE		8640			POTENCIA ACTIVA		2	
1100011	DT_DOUBLE		8640			POTENCIA REACTIVA		3	
1100012	DT_DOUBLE		8640			FACTOR POTENCIA		4	
1100013	DT_DOUBLE		8640			FRECUENCIA		5	

Tags 10.0_INVERSOR_1_1															
Channel	Datatype	Unit	Length	Minimum	Maximum	Description	NL_ArrayColu	Field	Medida	Tag_Aux1	Tag_Aux2	Tag_Aux3	Tag_Estacion	Tag_Familia	Tag_Sistema
Ref_Tempora	DT_DATE		8640			Ref_Temporal									
1100000	DT_DOUBLE		8640			STATUS		0 STATUS	INVERSOR_1_1				CT1	INVERSORES	1.1
1100009	DT_DOUBLE		8640			POTENCIA APARENTE		1 POTENCIA_APARENTE	INVERSOR_1_1				CT1	INVERSORES	1.1
1100010	DT_DOUBLE		8640			POTENCIA ACTIVA		2 POTENCIA_ACTIVA	INVERSOR_1_1				CT1	INVERSORES	1.1
1100011	DT_DOUBLE		8640			POTENCIA REACTIVA		3 POTENCIA_REACTIVA	INVERSOR_1_1				CT1	INVERSORES	1.1
1100012	DT_DOUBLE		8640			FACTOR POTENCIA		4 FACTOR_POTENCIA	INVERSOR_1_1				CT1	INVERSORES	1.1
1100013	DT_DOUBLE		8640			FRECUENCIA		5 FRECUENCIA	INVERSOR_1_1				CT1	INVERSORES	1.1

Figura 31 Antes y después de añadir los nuevos atributos

Si todo el proceso de sobre escritura en el archivo no ha generado ningún error, este se guarda en otro directorio donde se guardarán todos los archivos nuevos archivos con las modificaciones, permitiendo así mantener los archivos originales.

En caso de haber ocurrido algún error durante el proceso, se guardará el nombre del archivo en una lista para que, al finalizar el proceso de modificación, dicha lista sea mostrada al usuario y pueda solucionar la causa del error y volver a intentar el proceso.

5.4.2 Aplicación de migración

Una vez adaptados los archivos TDMS, con las etiquetas necesarias para el diseño de almacenamiento en InfluxDB, se desarrolla la aplicación que inserta los datos de los TDMS en InfluxDB. El nombre de esta aplicación es App_TDMS_to_Influx.

La parte inicial del desarrollo es muy similar a la mencionada en la aplicación anterior, se indica el directorio donde se encuentran los archivos TDMS que se desean insertar en InfluxDB, partiendo de la lista de archivos que se encuentran en el directorio se analiza archivo por archivo desglosándose por grupos y por canales de cada grupo. Llegados a cada canal, se obtienen todos los datos con las etiquetas y su referencia temporal. Estos datos se montan en un objeto como parámetro de entrada de la función de inserción desarrollada en un subvi.

La llamada a este subvi se realiza para cada uno de los grupos de cada archivo TDMS y el proceso de esta función es, a partir de los datos de entrada, es decir, los datos leídos del TDMS, diseñar la petición de escritura que se realiza a InfluxDB con el formato Line Protocol.

Se genera un array por cada campo, es decir, por cada field que se tenga en el objeto que recibe esta función. Y cada casilla del array contendrá la sentencia de Line Protocol para cada referencia temporal de ese field.

Este array de sentencias se pasa como parámetro a otro subvi encargado de realizar la petición HTTP mediante el método POST al servidor InfluxDB y pasando como cuerpo de la petición todas las sentencias formadas anteriormente.

Como en la aplicación explicada anteriormente, en caso de fallo en algún archivo TDMS, el nombre de este se almacenaría en una lista para corregir el error y volver a intentar el proceso.

5.4.3 Actualización de la aplicación general

Llegados a este punto, se dispone de los datos históricos en InfluxDB y es momento de preparar la aplicación principal para que inserte los datos obtenidos en tiempo real a la nueva base de datos InfluxDB.

Para ello solo será necesario modificar el módulo Registro de la aplicación actual, sustituyendo los Vis anteriores de escritura en TDMS por los nuevos. Se ha pensado un desarrollo de dos SubVIs que permita, en primer lugar, preparar los datos al formato propio de escritura Line Protocol y, en segundo lugar, de los datos formateados, realizar la escritura mediante petición POST al servidor de InfluxDB.

El primer SubVI, llamado Data_to_query.vi es el encargado de formatear los datos leídos y recibidos de los sensores en formato Line Protocol. El proceso que desarrolla empieza recibiendo los datos en un objeto donde la referencia es el instante temporal de esos datos, ya que la escritura se hace cada 10 segundos, todos los datos son referidos al mismo instante. Por cada referencia temporal existe un array de medidas o “_measurements” como se llaman en InfluxDB. Dentro de cada medida existe un array con un objeto por cada campo o “_field” como se llama en InfluxDB que contiene el nombre del campo, el valor de dicho campo y las etiquetas que hacen referencia a ese campo y medida.

Una vez se obtienen los datos de entrada, se debe analizar todo el objeto, variable por variable, para construir una cadena de texto que contenga todos los datos en el formato deseado. Recordando el formato Line Protocol, cuenta de 4 partes: el nombre de la medida, el grupo de clave-valor de las etiquetas, el grupo de clave-valor de las variables y por último, la referencia temporal en formato timestamp. Estas 4 partes se formatean en paralelo y posteriormente se concatenan en el orden correspondiente para dar formato a la query completa.

Una vez completado el proceso de preparación de la cadena de texto, se pasa como parámetro al siguiente SubVI que se encarga de enviar dicha cadena de texto al servidor. La comunicación se realiza mediante una petición HTTP POST como se ha visto en apartados anteriores.

5.5 Desarrollo de scripts de consulta con Flux.

Para las aplicaciones y funcionalidades vistas anteriormente, en concreto las que realizan consultas a la base de datos InfluxDB para obtener datos almacenados, deben incorporar en el body de la petición una consulta en lenguaje Flux que InfluxDB interpreta para identificar que parámetros o variables debe responder a dicha consulta.

Flux, además de filtrar los datos que se desean consultar, permite también realizar operaciones con los valores almacenados. En la web de la documentación oficial de Flux, se pueden encontrar multitud de operaciones como la media, el sumatorio incluso la integral de un grupo de valores. También se pueden realizar agrupaciones de datos para realizar operaciones obteniendo diferentes resultados para una misma variable.

En este apartado se presentan diferentes ejemplos de scripts para conocer mejor el lenguaje Flux y entender otras partes y funcionalidades del proyecto. Muchas de las consultas se realizan desde la parte web del proyecto para representar los datos, pero también se realizan diferentes consultas desde la aplicación desarrollada en LabVIEW para obtener nuevos datos a partir de otros realizando operaciones.

El primer ejemplo de consulta es para obtener datos de históricos referenciados entre dos fechas.

```
from(bucket: "Nombre_Bucket")
|> range(start: ${initialDate}, stop: ${finalDate})
|> filter(fn: (r) => ${ors})
|> window(period: ${resolution}s, createEmpty: true)
|> mean()
|> group(columns: ["Tag_ID"], mode: "by")
|> yield(name: "mean")
```

La consulta anterior se realiza desde el servidor desarrollado en NodeJS y se puede ver que tiene variables que se modifican con relación a los filtros que se reciben desde la web del proyecto. Una consulta siempre empieza filtrando por un rango temporal, en este caso, se puede ver que las fechas de inicio y fin del rango son aportadas por la web. Seguidamente se filtra por las etiquetas de los valores que se desea obtener, otro dato aportado por la web.

A continuación, se hace un inventariado y operación de los datos, primero se generan ventanas temporales de la resolución aportada por la web, seguidamente se realiza el cálculo de la media de los valores de cada ventana. Por último, agrupamos el resultado por la etiqueta ID para facilitar la representación en las gráficas de la web.

A continuación, se muestra un ejemplo de consultas con procedimientos y operaciones realizados en el mismo servidor de InfluxDB.

En el siguiente caso se desea obtener los valores de potencia a partir de los registros de tensión e intensidad almacenados cada 10 segundos. Para obtener el valor de potencia cada 10s se debe realizar el producto de los valores de intensidad y tensión en cada referencia temporal.

En la primera parte del script se realizan las dos consultas a los datos de tensión e intensidad, de esta forma se almacenan los datos en dos variables identificativas. Después del filtrado de los datos, se eliminan las columnas “_start”, “_stop” para las operaciones posteriores.

```
tension = from(bucket:
"Nombre_Bucket")
  |> range(start: %s , stop: %s
)
  |> filter(fn: (r) =>
r["Tag_Familia"] ==
"GATEWAY_STRINGS")
  |> filter(fn: (r) =>
r["Tag_ID"] == "%d")
  |> drop(columns: ["_start",
"_stop"])

intensidad = from(bucket:
"Nombre_Bucket")
  |> range(start: %s , stop: %s
)
  |> filter(fn: (r) =>
r["Tag_Familia"] ==
"GATEWAY_STRINGS")
  |> filter(fn: (r) =>
r["Tag_ID"] == "%d")
  |> drop(columns: ["_start",
"_stop"])
```

Se unen los datos de ambas variables obtenidas anteriormente con la columna “_time” como referencia conjunta. Con la función map() se genera otra tabla a partir de la anterior, para ello la función map() realiza la misma operación registro por registro. En este caso, asignamos los mismos valores a las etiquetas, el mismo instante temporal y, por último, el producto de la tensión e intensidad a la columna de los valores, obteniendo la potencia.

```
join( tables: {V: tension, I: intensidad}, on: ["_time"])
|> map(fn: (r) => ({
  Tag_Estacion: r.Tag_Estacion_V,
  Tag_Familia: r.Tag_Familia_V,
  Tag_Sistema: r.Tag_Sistema_V,
  _field: strings.split(v: r._field_I, t: "INTENSIDAD")[0] +
"POTENCIA",
  _measurement: "POTENCIAS",
  _time: r._time,
  _value: r._value_V * r._value_I
}))
|> yield(name: "result")
|> to( bucket: "POTENCIAS", org: "org", host:
"https://URL:8086", token: "token")
```

La última sentencia del script, es decir, la función to() realiza la escritura de la nueva tabla obtenida con los valores de la potencia en un nuevo bucket en la base de datos del parque en cuestión.

6 Implantación y testeo.

Cuando se tienen todos los desarrollos finalizados llega el momento de la puesta en marcha de todo el sistema.

En el caso de este proyecto, se tiene la ventaja de poder mantener en ejecución la versión anterior del sistema, esto permite realizar todas las pruebas de funcionamiento y la puesta en marcha sin el temor de perder información. Otro punto positivo es que se pueden realizar comparaciones de los datos proporcionados por ambas versiones permitiendo verificar el correcto funcionamiento del nuevo sistema.

Los pasos a seguir para la puesta en marcha son:

- Ejecución del servidor de InfluxDB en su dirección de producción y comprobación de su correcto funcionamiento mediante pruebas de acceso, escritura y lectura de datos ficticios.
- Ejecución de la nueva versión de la aplicación de lectura de datos de los sensores e inversores con escritura de los datos en el servidor de InfluxDB de producción.
- Comprobación de los datos almacenados en tiempo real en el servidor de la base de datos mediante la web cliente que proporciona el propio servicio de InfluxDB.
- Ejecución de las nuevas versiones del servidor REST y la web cliente para comprobar su funcionamiento con los datos almacenados en tiempo real.
- Comparación de datos entre la antigua y la nueva versión del sistema.



Figura 32 Sección de gráficos de históricos

Después de este primer proceso de puesta en marcha, y una vez se verifica que todo funciona correctamente, se realiza la puesta en marcha de las aplicaciones de recuperación de datos históricos hasta la fecha.

- Ejecución de la aplicación para añadir los nuevos atributos a los datos almacenados en los archivos TDMS.
- Ejecución de la aplicación de escritura de los datos almacenados en los archivos TDMS en la base de datos InfluxDB.
- Comprobación de los datos de históricos almacenados en el servidor de la base de datos mediante la web cliente que proporciona el propio servicio de InfluxDB.
- Comparación de datos entre la antigua y la nueva versión del sistema.

A continuación, se muestra varias imágenes obtenidas de la propia web donde se muestran las diferentes formas de monitorizar los datos. En todas las gráficas con datos históricos se ofrece la opción de realizar un filtrado por fecha y ajustar la resolución de los datos a mostrar entre diferentes opciones. También se ha desarrollado la funcionalidad de exportar dichos datos en diferentes archivos .csv, un archivo por gráfica.



Figura 33 Sección de gráficos de históricos

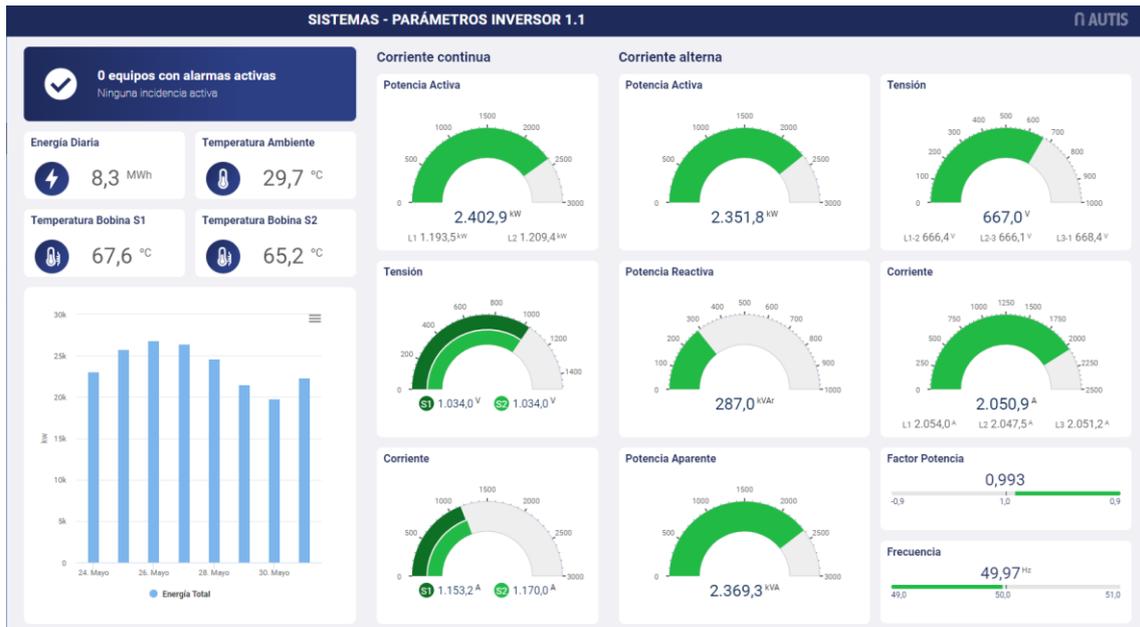


Figura 34 Sección de gráficos de inversores



Figura 35 Historiales diarios de energía generada

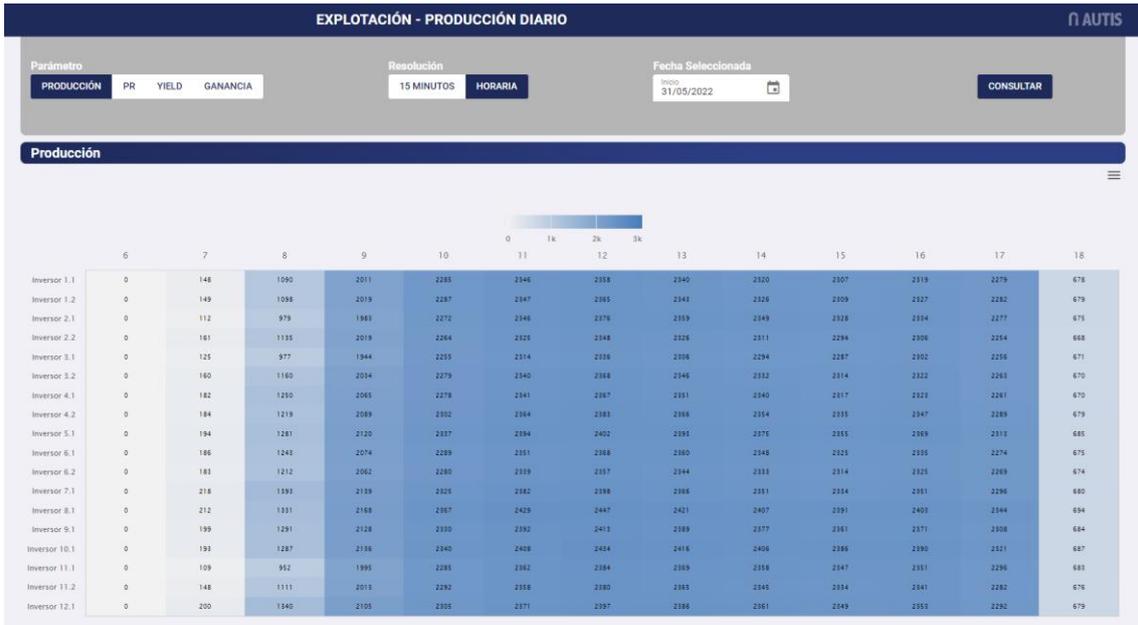


Figura 36 Mapa de calor de los datos de producción diaria

7 Conclusiones y trabajo futuro

En este proyecto se han unido diferentes tecnologías y se han implementado diferentes funcionalidades donde la variedad de tecnologías trabaja en conjunto.

Después del desarrollo realizado se puede decir que los objetivos definidos se han cumplido, centralizando los datos y evitando posibles pérdidas de información. Otra mejora sustancial es la disminución del tiempo de acceso y representación de los datos en la web cliente. Se confirma que InfluxDB mejora los tiempos de procesamiento y lectura de datos en algunas consultas de hasta un 75%.

Se ha llegado al momento donde se realizan los trabajos de mantenimiento y actualizaciones sobre el producto.

Esta implementación se ha desarrollado en concreto para un parque fotovoltaico, pero ahora se pretende expandir el cambio llevado a cabo en todos los demás parques que disponen del producto PSFVIEW.

Por otra parte, se está desarrollando una librería propia para el manejo de InfluxDB sobre LabVIEW, esto permitirá implementar todo el potencial de InfluxDB en proyectos de diferentes características y facilitar el aprendizaje y uso de esta nueva base de datos en el entorno de la empresa.

Otra vista de futuro es la implementación de InfluxDB para la monitorización de equipos y Workstation con la aplicación Telegraf, desarrollada por InfluxData. Esta aplicación permite instalarse como un servicio en segundo plano y monitorizar características del hardware, red y otros protocolos de comunicación como MQTT del propio equipo, donde todos los datos son enviados a una base de datos InfluxDB.

Estas implementaciones permiten monitorizar y controlar, en proyectos grandes, las necesidades de hardware y comunicaciones para una posible mejora o actualización.

Puesto que este documento es un trabajo final de grado, el objetivo principal es el aprendizaje, extrayéndose del ámbito específico de las tecnologías



8 Bibliografía

InfluxDATA. InfluxDB Documentation. Disponible en:

[HTTPS://DOCS.INFLUXDATA.COM/INFLUXDB/V2.1/GET-STARTED/](https://docs.influxdata.com/influxdb/v2.1/get-started/)

InfluxDATA. Flux Documentation. Disponible en:

[HTTPS://DOCS.INFLUXDATA.COM/FLUX/V0.X/GET-STARTED/](https://docs.influxdata.com/flux/v0.x/get-started/)

InfluxDATA. InfluxDB API v2.1. Disponible en:

[HTTPS://DOCS.INFLUXDATA.COM/INFLUXDB/V2.1/API/#TAG/QUICK-START](https://docs.influxdata.com/influxdb/v2.1/api/#tag/quick-start)

InfluxDATA. Time to awesome. Disponible en:

[HTTPS://AWESOME.INFLUXDATA.COM/DOCS/PART-1](https://awesome.influxdata.com/docs/part-1)

Oracle. ¿Qué es el big data? Disponible en:

[HTTPS://WWW.ORACLE.COM/ES/BIG-DATA/WHAT-IS-BIG-DATA/](https://www.oracle.com/es/big-data/what-is-big-data/)

Deloitte. ¿Qué es la industria 4.0?. Disponible en:

[HTTPS://WWW2.DELOITTE.COM/ES/ES/PAGES/MANUFACTURING/ARTICLES/QUE-ES-LA-INDUSTRIA-4.0.HTML](https://www2.deloitte.com/es/es/pages/manufacturing/articles/que-es-la-industria-4.0.html)

Deloitte. IoT-Internet of things. Disponible en:

[HTTPS://WWW2.DELOITTE.COM/ES/ES/PAGES/TECHNOLOGY/ARTICLES/IOT-INTERNET-OF-THINGS.HTML](https://www2.deloitte.com/es/es/pages/technology/articles/iot-internet-of-things.html)

National Instruments. El formato NI-TDMS. Fecha 25 mayo 2022. Disponible en:

[HTTPS://WWW.NI.COM/ES-ES/SUPPORT/DOCUMENTATION/SUPPLEMENTAL/06/THE-NI-TDMS-FILE-FORMAT.HTML](https://www.ni.com/es-es/support/documentation/supplemental/06/the-ni-tdms-file-format.html)

National Instruments. ¿Qué es LabVIEW?. Disponible en:

[HTTPS://WWW.NI.COM/ES-ES/SHOP/LABVIEW.HTML](https://www.ni.com/es-es/shop/labview.html)

OpenJSFoundation. Acerca de Node.js. Disponible es:

[HTTPS://NODEJS.ORG/ES/ABOUT/](https://nodejs.org/es/about/)

